

**VŠB - Technická univerzita Ostrava**  
**Fakulta elektrotechniky a informatiky**

**DIPLOMOVÁ PRÁCE**

**VŠB - Technická univerzita Ostrava**  
**Fakulta elektrotechniky a informatiky**  
**Katedra Elektroniky**

**Návrh a realizace řídicího systému pro napájecí jednotku  
krokových motorků**

Control System for Power Supply Unit of Stepper Motors - Design and  
Realization

## Zadání diplomové práce

Student:

**Bc. Miloslav Tměj**

Studijní program:

N2649 Elektrotechnika

Studijní obor:

2612T015 Elektronika

Téma:

Návrh a realizace řídicího systému pro napájecí jednotku krokových motorků.

Control System for Power Supply Unit of Stepper Motors - Design and Realization.

Zásady pro vypracování:

1. Specifikujte požadavky na řídicí systémy dvoufázových krokových motorků.
2. Rozeberte možnosti komunikace, řízení a ovládání výkonové části měničů pro krokové motorky.
3. Na základě pokynů vedoucího diplomové práce navrhnete řídicí systém pro vybranou řídicí jednotku s možností autonomního ovládání bez přítomnosti PC.

Seznam doporučené odborné literatury:

Firemní literatura RMS Technologies ([www.rmsmotion.com/](http://www.rmsmotion.com/))

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Ing. Václav Sládeček, Ph.D.**

Datum zadání: 01.09.2013

Datum odevzdání: 07.05.2014



doc. Ing. Petr Palacký, Ph.D.  
vedoucí katedry




prof. RNDr. Václav Snášel, CSc.  
děkan fakulty

### **Prohlášení studenta:**

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě dne: .....2.5. 2016.....

Podpis..........

**Poděkování:**

Děkuji svému vedoucímu diplomové práce, kterým byl pan Ing. Václav Sládeček, Ph.D., za vhodnou metodickou a odbornou pomoc při zpracování této diplomové práce.

## **Abstrakt:**

Zadání této diplomové práce vycházelo z potřeby vytvořit pomůcku pro nastavování ovladače krokových motorků. V práci je popsáno základní rozdělení krokových motorků, jejich princip a způsoby ovládání krokových motorků. V další části práce jsou popsány ovladače pro krokové motorky a druhy komunikací s těmito ovladači. Praktická část se zabývá realizací navrhovaného zařízení. Je zde popsána konstrukce a elektrické zapojení navrhovaného zařízení. Nechybí ani popis hlavních součástí, které jsou v zapojení použity. Popsána je i jejich vzájemná komunikace. Poslední část práce se zabývá tvorbou programu pro mikrokontrolér.

## **Abstract:**

The requirements of this master thesis were based on the need to create a tool for setting up a controller of stepper motors. The thesis describes the basic classification of stepper motors, their basic principles, and the methods of controlling of stepper motors. The following part of the work describes the controllers of stepper motors and types of communication with the controllers. The practical part deals with implementation of a device projected. It includes description of the structure and electrical connection of the device projected. It also contains description of the main parts that were used within the connection and their mutual communication. The last part of the thesis deals with creation of the microcontroller program.

## **Klíčová slova:**

Krokový motor, úhel kroku, krok, řídicí impulsy, RS-485, mikrokontrolér, ovladač R356

## **Key words:**

Stepper motor, step angle, step, control impulses, RS-485, microcontroller, controller R356

## Seznam použitých symbolů a zkratek:

$f$	Frekvence	[Hz]
$i$	Okamžitá hodnota proudu	[A]
$I$	Proud	[A]
$M$	Točivý moment	[Nm]
$P$	Výkon	[W]
$p$	Okamžitý výkon	[W]
$R_m$	Magnetický odpor (Reluktance)	[H <sup>-1</sup> ]
$t$	Čas	[s]
$T$	Doba	[s]
$u$	Okamžitá hodnota napětí	[V]
$U$	Napětí	[V]
$\Delta\Phi$	Velikost kroku	[°, rad]
$\Phi$	Magnetický tok	[Wb]
$\vartheta_j$	Teplota čipu	[K]
$\vartheta_c$	Teplota pouzdra	[K]
$\vartheta_s$	Teplota chladiče	[K]
$\vartheta_a$	Teplota okolí	[K]
$R_{th(j-c)}$	Vnitřní tepelný odpor (čip - pouzdro)	[K/W]
$R_{th(c-s)}$	Tepelný odpor pouzdro - chladič	[K/W]
$R_{th(s-a)}$	Tepelný odpor chladič – okolí	[K/W]
AC	Střídavý proud (Alternating Current)	
ACK	Acknowledge character potvrzovací znak	
ASCII	American Standard Code for Information Interchange americký standardní kód pro výměnu informací	
CAN bus	Controller Area Network datová sběrnice pro průmyslové prostředí (vozidla)	
CCW	Counterclockwise otáčení proti směru hodinových ručiček	
CNC	Computer Numeric Control číslicově řízený stroj	
CRC	Cyclic Redundancy Check cyklický redundantní kód, kontrolní součet	
CW	Clockwise otáčení ve směru hodinových ručiček	
DC	Stejnoseměrný proud (Direct current)	
EEPROM	Electrically Erasable Programmable Read-Only Memory je elektricky mazatelná nevolatilní paměť	
FLASH	Nevolatilní elektricky programovatelná paměť s libovolným přístupem	
FRAM	Ferroelectric Random Access Memory je nevolatilní paměť s přímým přístupem	
IEC	Označení mezinárodního souboru elektrotechnických norem (International Electrotechnical Commission)	
IIC, I <sup>2</sup> C	Inter-Integrated Circuit typ komunikačního protokolu sériového rozhraní	
IGBT	Insulated Gate Bipolar Transistor IGBT bipolární tranzistor s izolovaným hradlem	
LCD	Liquid crystal display je displej z tekutých krystalů	
MOSFET	Metal Oxide Semiconductor Field Effect Transistor je tranzistor řízený elektrickým polem	
MODBUS	Komunikační protokol	
NEMA	National Electrical Manufacturers Association je sdružení se sídlem v USA, které tvoří normy pro elektrická a lékařská zařízení	
PWM	Pulsně šířková modulace	
RS485	Standard sériové komunikace podle EIA485	
SPI	Serial Peripheral Interface sériová komunikace	
TFT	Thin-Film Transistors – typ LCD zobrazovače	
VR	Variable Reluctance – proměnná reluktance, proměnný magnetický odpor	

## Obsah

1.	Úvod	1
2.	Krokové motory	2
2.1.	Historie	2
2.2.	Funkce, princip a rozdělení krokových motorů	3
2.2.1	Krokový motor s pasívním rotorem	3
2.2.2	Krokový motor s permanentním magnetem	4
2.2.3	Hybridní krokový motor	4
	Využití krokových motorů	6
2.3.	Způsoby řízení krokových motorů	7
2.3.1	Unipolární řízení krokových motorků	7
2.3.2	Bipolární řízení krokových motorků	9
2.3.3	Mikrokrokování	10
2.3.4	Statická momentová charakteristika	12
2.3.4	Dynamická momentová charakteristika	12
3.	Komunikační protokoly	14
3.1.	Řízení pomocí pulsů, směru a vypnutí „disable“	14
3.2.	Řízení pomocí sběrnice CAN BUS	15
3.3.	Řízení pomocí sběrnice RS485	17
3.4.	Použití sběrnice RS485 pro řízení krokových motorků	19
3.4.1.	Kontroléry s procesorem M1486	19
3.4.2.	Kontrolér R356 od firmy RMS Technologies Inc	21
3.4.3.	Ovládání kontroléru R356	23
4.	Realizace	25
4.1.	Rozbor zadání	25
4.2.	Volba procesoru	26
4.3.	Volba displeje	28
4.4.	Popis obvodu FT800	30
4.5.	Paměť EEPROM (FRAM)	33
4.6.	Zdrojová část	35
4.7.	Konstrukce přístroje	37
4.8.	Schéma zapojení	38



4.9. Návrh plošného spoje .....	40
5. Program .....	42
5.1. Zadání úkolu.....	42
5.2. Analýza projektu .....	42
5.3. Výběr programovacího jazyka a vývojového prostředí .....	45
5.4. Psaní programu.....	46
5.4.1. Popis hlavní části programu.....	47
5.4.2. Popis programu vykonaného při přerušení .....	48
5.4.3. Hlavní podprogramy .....	49
5.5. Ladění aplikace a oživení desky plošného spoje.....	52
5.6. Testování aplikace.....	55
5.7. Parametry výrobku .....	55
6. Závěr .....	56
Bibliografie .....	58

## Seznam obrázků:

Obr. 1 Jeden z patentů zabývající se krokovým motorem [2]	2
Obr. 3 Princip krokového motoru s pasívním rotorem [4]	3
Obr. 2 Rozdělení krokových motorů podle konstrukce	3
Obr. 4 Základní princip krokového motoru s aktivním rotorem [4]	4
Obr. 5 Základní princip hybridního krokového motoru [5]	4
Obr. 6 Řez hybridním krokovým motorem [7]	4
Obr. 7 Dvoufázový hybridní krokový motor podle standardu NEMA [7]	6
Obr. 8 Příklady využití krokových motorků [8]	6
Obr. 9 Rozdíl mezi bipolárním a unipolárním řízením	7
Obr. 10 Základní princip řízení motoru s pasívním rotorem [9]	7
Obr. 11 Čtyřtaktní unipolární řízení krokového motoru s pasívním rotorem s magnetizací jedné fáze	8
Obr. 12 Princip zvýšení počtu kroků reluktančního krokového motoru zvýšením počtu zubů na pólech	8
Obr. 13 Osmíktaktní unipolární řízení krokového motoru s pasívním rotorem s magnetizací dvou fází	8
Obr. 14 Bipolární řízení krokových motorků s aktivním rotorem [10]	9
Obr. 15 Různé druhy zapojení krokového motorku [11]	9
Obr. 16 Vektorový diagram pootočení	10
Obr. 17 Různé metody pro zvýšení počtu kroků dvoufázového hybridního krokového motoru [13]	11
Obr. 18 Statická momentová charakteristika krokového motoru [14]	12
Obr. 19 Dynamická momentová charakteristika krokového motoru [14]	13
Obr. 21 Průběh momentové charakteristiky krokového motoru s poklesy vlivem rezonancí [15]	13
Obr. 20 Odezva na jeden krok [14]	13
Obr. 22 Ovládání krokových motorků - nadřazená jednotka, ovladač, motor a komunikace mezi nimi	14
Obr. 23 Příklad ovládání ovladače pro krokový motor pomocí pulsů	14
Obr. 24 Příklad zapojení komunikace CAN	15
Obr. 25 Přenášena zpráva po sběrnici protokolem CAN 2.0A	16
Obr. 26 Příklad ovladače krokových motorků se sběrnici CAN 2.0A	17
Obr. 27 Topologie a příklad zapojení sítě RS485 [17]	17
Obr. 28 Rámec přenosu jednoho byte informace [18]	18
Obr. 29 Minimální úrovně pro sběrnici RS485 [17]	18
Obr. 30 Full-duplex sběrnice RS422 a Half-duplex sběrnice RS485 (vpravo) [17]	18
Obr. 31 Programovatelná jednotka typu CD a krokový motor od firmy Microcon [19]	20
Obr. 32 Kontrolér R356 od firmy RMS Technologies Inc [20]	21
Obr. 33 Rozmístění konektorů na kontroléru R356 [20]	21
Obr. 34 Zapojení několika driverů R356 v síti [20]	22
Obr. 35 Předběžný návrh nového zařízení	25
Obr. 36 Předběžné blokové schéma zobrazující základní části navrhovaného zařízení	26
Obr. 37 Struktura procesoru AT32UC3B1256 [18]	27
Obr. 38 Struktura paměti a umístění bootloaderu v procesoru AT32UC3B1256 [18]	28
Obr. 39 Příklad zadávání ASCII řetězce pomocí křížového ovladače	28
Obr. 40 Základní blokové schéma integrovaného obvodu FT800 [21]	29
Obr. 41 Vývojová deska WM800C35A-D [21]	29
Obr. 42 Blokové schéma integrovaného obvodu FT800	30
Obr. 44 Signály generované pro displej [22]	31
Obr. 43 SPI komunikace [30]	31
Obr. 45 Příklady některých „widgetů“, které nabízí FT800 [21]	32
Obr. 46 Základní princip paměti FRAM [23]	33
Obr. 47 Adresování použité paměti [25]	34
Obr. 48 IIC komunikace s použitím HW adresy [24]	34
Obr. 49 IIC komunikace startbit a stopbit [25]	34
Obr. 50 Tepelné schéma stabilizátoru	36
Obr. 51 Rozměry použité krabičky [26]	37
Obr. 52 Konstrukční uspořádání výrobku	37
Obr. 53 Upravené blokové schéma navrhovaného zařízení	38
Obr. 54 Zapojení integrovaného obvodu FT800 [21]	39

<i>Obr. 55 Zapojení zesilovače zvuku [21]</i> .....	39
<i>Obr. 56 Původní návrh jednostranné desky s propoji (transparentní pohled ze strany TOP)</i> .....	40
<i>Obr. 57 Vyrobená deska plošných spojů (obě strany)</i> .....	41
<i>Obr. 58 Tři základní části zařízení</i> .....	42
<i>Obr. 59 Minimální délka jednoho byte při rychlosti 38,4kBd</i> .....	43
<i>Obr. 60 Rozložení ovládacích prvků na displeji</i> .....	44
<i>Obr. 61 Vývojové prostředí AVR32 studio [18]</i> .....	45
<i>Obr. 62 Vývojový diagram pro hlavní část programu</i> .....	46
<i>Obr. 63 Vývojový diagram pro program vykonávaný v přerušení</i> .....	48
<i>Obr. 64 Podprogram „Test dotykového panelu“</i> .....	50
<i>Obr. 65 Konstrukce obrazu a jeho odeslání na displej</i> .....	51
<i>Obr. 66 Vývojový diagram podprogramu pro vysílání</i> .....	51
<i>Obr. 67 Test přerušení procesoru</i> .....	52
<i>Obr. 68 Kontrola přepnutí převodníku RS485 po odvysílání při různých rychlostech</i> .....	52
<i>Obr. 69 Spínaný zdroj pro podsvětlení displeje [21]</i> .....	53
<i>Obr. 70 Zobrazení odeslaných a přijatých dat</i> .....	54
<i>Obr. 71 Ovládání kurzoru</i> .....	54
<i>Obr. 72 Testování zařízení</i> .....	55

## Seznam tabulek:

<i>Tab. 1 Tabulka významu jednotlivých znaků standardu NEMA [6]</i> .....	5
<i>Tab. 2 Popis zprávy sběrnice CAN [16]</i> .....	16
<i>Tab. 3 Příklad použitých příkazů kontroléru M1486 [19]</i> .....	20
<i>Tab. 4 Zapojení konektoru cannon 15 [20]</i> .....	22
<i>Tab. 5 Formát odesílaných dat do kontroléru [20]</i> .....	23
<i>Tab. 6 Přiřazení ASCII znaků k adresám [20]</i> .....	23
<i>Tab. 7 Přiřazení ASCII znaků pro společné ovládání více zařízení najednou [20]</i> .....	23
<i>Tab. 8 Rozdělení příkazů [20]</i> .....	24
<i>Tab. 9 Odpověď na příkaz [20]</i> .....	24
<i>Tab. 10 Rozdělení adresového prostoru integrovaného obvodu FT800</i> .....	32
<i>Tab. 11 Tabulka napěťových úrovní na navrhované desce</i> .....	35
<i>Tab. 12 Tabulka přibližného odběru součástek</i> .....	35
<i>Tab. 13 Informativní měření teploty mezi chladičem a integrovaným obvodem (ustálení 20min.)</i> .....	54

## 1. Úvod

Během posledních několika století bylo vynalezeno velké množství různých pohonů, které pracují na různých principech. Většina z nich našla v poměrně krátké době svoje využití.

Jinak to bylo u krokového motoru. Ten byl vyvinut na základě synchronního motoru. Dlouhou dobu však trvalo, než si našel větší uplatnění. Hlavním důvodem je skutečnost, že krokový motor potřebuje ke své činnosti spínací elektroniku. Teprve s rozmachem tranzistorů a jiných polovodiivých spínacích prvků se začaly krokové motory uplatňovat. V současné době nás tyto motory doslova obklopují. Jsou součástí automobilů, klimatizace, topení, průmyslové, ale i domácí elektroniky. Velké uplatnění našly v paměťových médiích. To byl asi hlavní důvod toho, že koncem minulého století svět investoval do vývoje krokových motorků a jejich ovladačů obrovské prostředky.

Důvodem pro použití krokových motorků je jejich přesnost a hlavně dlouhodobá spolehlivost. Spolehlivosti přispívá i to, že pro dosažení přesné polohy nejsou v těchto pohonech nutné dodatečné snímače. Další dobře využitelnou vlastností je klidový moment těchto motorů. I v klidu si udrží svoji polohu velmi přesně s relativně velkým momentem, aniž by došlo k jejich poškození.

V současné době se v paměťových médiích krokové motorky pomalu přestávají používat. Jejich popularita však neklesá a díky svým vlastnostem nacházejí stále větší uplatnění ve výrobních zařízeních. Přesný, bezchybný pohyb je zde základem pro výrobu kvalitních výrobků.

Rychlý rozvoj a zkvalitnění elektronických součástek posunuje hranice pro použití krokových motorů. Elektronický ovladač se stává nedílnou součástí krokového motoru. Uživatel se již nestará o jednotlivá sepnutí výkonových prvků, ale komunikuje s „motorem“ na vyšší úrovni. Ovládání pomocí pulsů není žádnou novinkou. V posledních letech se však objevily ovladače, které je možné ovládat pomocí některého z průmyslových komunikačních protokolů. Komunikace probíhá nejčastěji komunikačním protokolem RS485 (popřípadě CAN, atd.). Řídící jednotka potom neposílá do jednotlivých driverů pulsy a informaci o směru, ale posílá přímo příkazy s hodnotou nové polohy, dálkově nastavuje proud, rychlost, zrychlení a podobně. Výhodou tohoto způsobu komunikace je skutečnost, že je možné pomocí kontrolních součtů odstranit chyby způsobené rušením linky.

Tato diplomová práce se zabývá mimo jiné kontrolérem R356, který ovládá krokové motorky a je řízen pomocí linky RS485. Kontrolér vyrábí firma RMS Technologies, která patří mezi světově uznávané výrobce průmyslových zařízení pro ovládání krokových motorů. Aby bylo možné pochopit činnost ovladače, jsou v první části práce popsány krokové motory. Jejich principy, rozdělení, momentová charakteristika, základní krokování a princip rozdělení základního kroku.

Dále jsou zde popsány základní druhy komunikačních protokolů používaných pro ovládání průmyslových ovladačů a kontrolérů.

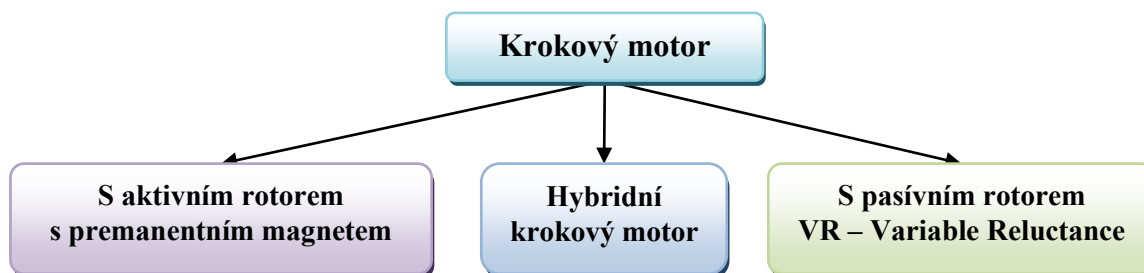
Druhá část diplomové práce je věnována realizaci zařízení, které má nahradit počítač při zadávání parametrů do kontroléru. Tato část diplomové práce se zabývá návrhem elektrického obvodu a konstrukcí zařízení. V poslední části je pak popsán program do jednočipového procesoru, který je základem celé aplikace.



## 2.2. Funkce, princip a rozdělení krokových motorů

Krokový motor se od ostatních elektrických motorů liší tím, že má každou svoji otáčku rozdělenou na jednotlivé úseky. Tyto úseky se potom nazývají kroky a jejich velikost je udávána ve stupních, popřípadě radiánech. Využitím této vlastnosti dostáváme pohon, který je možné řídit po jednotlivých krocích bez nutnosti používat drahé snímače, které často zvyšují poruchovost. Navíc jako další vlastnost je využita jeho schopnost vytvořit plný točivý moment i v klidové poloze.

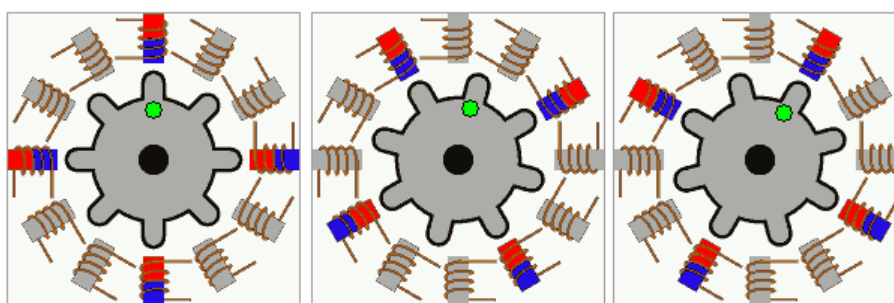
Krokové motory se dají rozdělit nejvýrazněji podle konstrukce. Dále je možné dělit krokové motory podle počtu fází, počtu pólů, počtu rotorů a statorů, podle napájecího napětí atd.



Obr. 2 Rozdělení krokových motorů podle konstrukce

### 2.2.1 Krokový motor s pasívním rotorem

Jak již bylo zmíněno v kapitole o historii, byl tento typ používán jako první. Tento motor se také nazývá „VR – Variable Reluctance“, tedy motor s proměnnou reluktancí. Z názvu vyplývá, že motor pracuje na reluktančním principu. Zjednodušeně řečeno, rotor si hledá takovou polohu, aby magnetický tok měl co nejmenší magnetický odpor. Řízení takového motoru se potom provádí pomocí přepínání cívek statoru, jak je vidět na velmi zjednodušeném obrázku. Protilehlé cívky statoru jsou zapojeny do párů. Rotor i stator mají různý počet pólových nástavců a jsou tvořeny většinou z magneticky měkkých ocelových plechů. Vzduchová mezera mezi póly rotoru a statoru je velmi malá mezi 0,2 až 0,02mm, což souvisí s momentem. Tento typ motorů však dosahuje obecně menších momentů než motory s permanentním magnetem a také nemají tzv. aretační moment. To je, že pokud neprotéká cívkami proud, nedrží si polohu. Velikost kroku bývá u tohoto typu motoru v běžné mezi pěti až patnácti stupni. Jeho výhodou jsou nízké náklady na výrobu a velmi nízká poruchovost. [3]

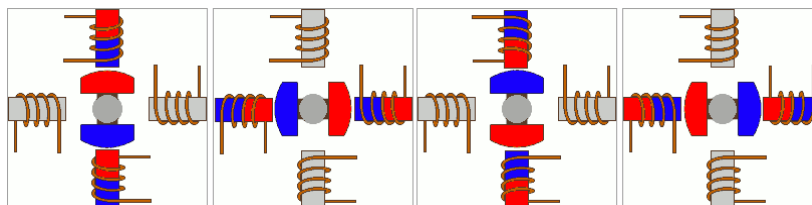


Obr. 3 Princip krokového motoru s pasívním rotorem [4]

Krokový motor s pasívním rotorem je vyráběn v různých provedeních. Liší se počtem fází ale i polohou cívek vůči rotoru a podobně. Vyrábí se i lineární verze. Je možné ho výrobně přizpůsobit dané aplikaci. Pro zjemnění kroku se používají jemné nástavce (zuby), které se na jednotlivých pólech nekryjí. Rozdělení kroků bude ještě popsáno v dalších kapitolách.

### 2.2.2 Krokový motor s permanentním magnetem

Další typ krokového motoru má rotor vyrobený z permanentního magnetu. Je také známý pod názvem „krokový motor s aktivním rotorem“. Princip pro jeho rozhýbání je velmi jednoduchý. Jelikož se dva opačné magnetické póly přitahují a souhlasně odpuzují, stačí vytvořit správným přepínáním cívek magnetické pole na straně statoru a jeho pohybem nebo zastavením ovládat rotor.

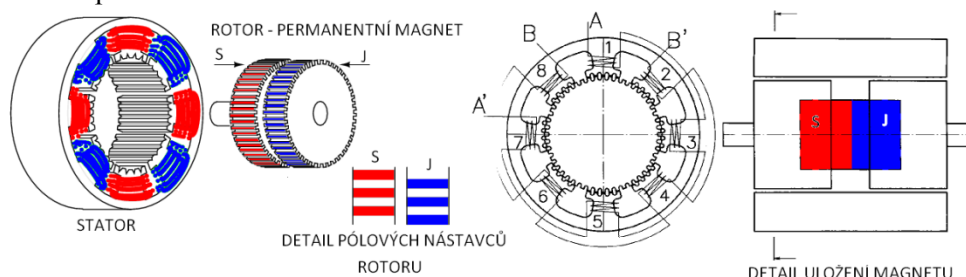


Obr. 4 Základní princip krokového motoru s aktivním rotorem [4]

Skutečný dvojfázový krokový motor má potom dvojnásobný počet pólů statoru oproti rotoru. Na rotoru se střádají magnetické póly. Dosahovaný úhel kroku u těchto motorů bývá  $7,5^\circ$ .

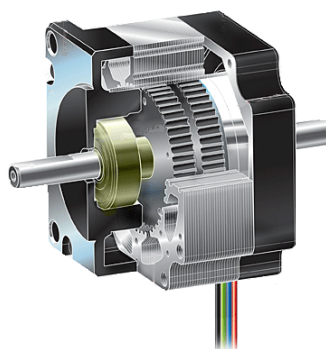
### 2.2.3 Hybridní krokový motor

Spojením předchozích dvou typů krokových motorů vznikl takzvaný hybridní krokový motor. Hybridní motory dosahují oproti motorům s pasivním rotorem vyšších hodnot momentu. Někteří výrobci udávají, že hybridní motory a motory s permanentním magnetem mají až o 40% vyšší moment než srovnatelné krokové motory s pasivním rotorem. Vyšší moment je však vyvážen složitější konstrukcí. To se pak odrazí na ceně.



Obr. 5 Základní princip hybridního krokového motoru [5]

Pokud mluvíme o hybridních krokových motorech, je nutné zmínit, že právě dvoufázové hybridní krokové motory jsou nejvíce vyráběným a používaným druhem krokových motorů. Je to díky tomu, že mají velký moment a malý základní krok.



Obr. 6 Řez hybridním krokovým motorem [7]

Hybridní krokový motor je složený klasicky z rotoru a statoru. Rotor je tvořen kvalitním permanentním magnetem, který je uložený podél osy. Na rotoru jsou dva ozubené nástavce válcovitého tvaru. Tyto nástavce mají shodný počet zubů, jsou však oproti sobě posunuty o půl zubu. Díky permanentnímu magnetu má každý z těchto ozubených nástavců jiný magnetický pól. Jeden jižní



a druhý severní. Je tím docíleno toho, že dva po sobě jdoucí póly rotoru mají každý jiný magnetický pól. Stator je tvořen (v našem případě) osmicí cívek na pólových nástavcích, které jsou pospojovány do dvou fází. Každý z pólových nástavců je zakončen na straně směrem k rotoru pěticí zubů.

Hybridní krokové motory se mohou lišit podle výrobce nejen svými parametry, ale i konstrukcí. Některé motory jsou vyráběny na míru pro použitou aplikaci, některé mají univerzální tvar. Postupem času se snažili výrobci sjednotit tvar i velikosti přírub tak, aby bylo možné tyto motory vzájemně zaměňovat a používat různé typy od různých výrobců. Tento trend je vlastně patrný i u jiných typů motorů jako jsou například motory asynchronní.

Jedním ze standardů, který určuje tvary krokových motorů je standard NEMA. Tento standard patří pod „The US-based National Electrical Manufacturers Association“.

Motory od různých výrobců jsou díky tomuto standardu záměnné a jejich konstrukce jsou si podobné. Rozdíly mezi motory jsou v dosahovaném momentu, rychlosti, účinnosti, v mechanických vlastnostech, ale také v životnosti motorů. Renomovaní výrobci se snaží právě v těchto vlastnostech vyniknout. Stává se, že motory od různých výrobců mají při stejných rozměrech rozdílný moment, rozdílné oteplení a vydávají rozdílný hluk.

Vraťme se však ke zmíněnému standardu. Z plného označení motoru je možné vyčíst následující údaje. [6]

"NEMA" DDMMLLL-CCCIVVVSSSW

Tab. 1 Tabulka významu jednotlivých znaků standardu NEMA [6]

Znak	Hodnota	Jednotka
DD	Průměr/Velikost příruby	palců · 10
MM	Montážní typ	C: Příruba s drážkami; D: Čelo se závitovými otvory; CD: Příruba s otvory
LLL	Délka	palců · 10
CCC	Proud fáze	ampér · 10
I	Izolační třída	Maximální pracovní teplota: A: 105 °C; B: 108 °C; F: 155 °C; H: 180 °C Třída B používaná pro 60Hz motory (USA) Třída F používaná pro 50Hz motory. Obecně řečeno, teplota 5°C nad maximální dovolenou teplotu snižuje životnost motoru na polovinu.
VVV	Rozsah fázového napětí	Volty · 10
SSS	Kroky	Kroků na jednu otáčku
W	Kód vinutí	Počet vodičů pro připojení: A: 2 vodiče; B: 3 vodiče; C: 4 vodiče; D: 5 vodiče; E: 6 vodiče; F: 8 vodiče

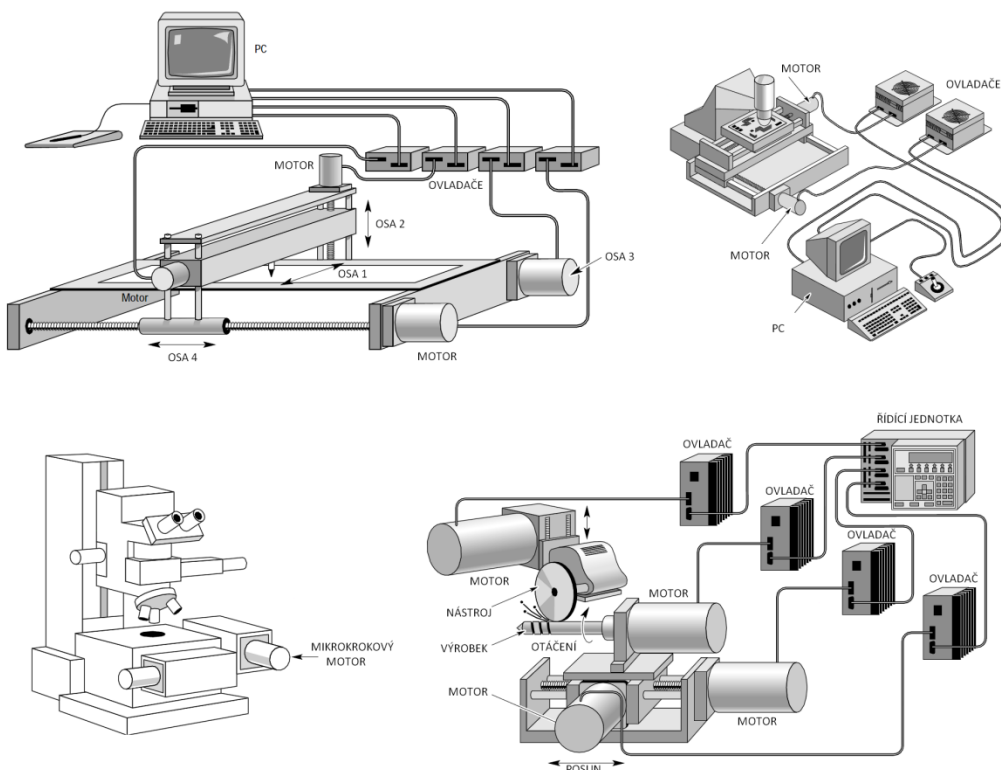
Hybridní motory jsou vhodné pro vysoké rozlišení (200-500 kroků na otáčku) a vyskytují se v mnoha výkonových třídách až do 3 kW. V průmyslových aplikacích jsou mezi krokovými motory hybridní motory zcela dominantní.



Obr. 7 Dvoufázový hybridní krokový motor podle standardu NEMA [7]

### Využití krokových motorů

Vlastnosti krokových motorů předurčují jejich využití v aplikacích, kde se vyžaduje přesný pohyb a přesné zachování klidové polohy. Paměťová technika spotřebovala kvanta krokových motorků, nyní se jich však pomalu zbavuje. Již delší čas se tento typ motorků využívá v CNC strojích, v plotterech, ale i k přesnému natáčení dalekohledů, antén a kamer. Krokové motory také nalézají stále větší uplatnění v lékařských oborech.

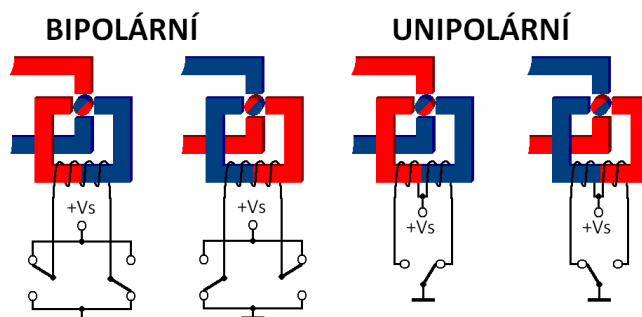


Obr. 8 Příklady využití krokových motorků [8]

## 2.3. Způsoby řízení krokových motorů

Podle způsobu napájení jednotlivých fází vinutí je možné řízení krokového motoru rozdělit následovně:

- **Unipolární řízení** – toto řízení je používáno hlavně u krokových motorů bez permanentního magnetu. Na vinutí je přiváděno napájení pouze jedné polaridy.
- **Bipolární řízení** – uplatňuje se pouze u motorů, kde je v rotoru použit permanentní magnet. Přivedením různých polarit na vinutí fáze umožňuje zvýšit počet kroků.



Obr. 9 Rozdíl mezi bipolárním a unipolárním řízením

Řízení krokového motoru je možné dále rozdělit podle počtu kroků na jednu periodu:

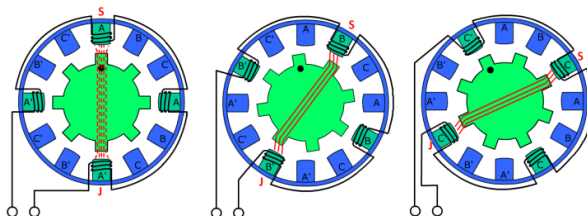
- Čtyřtaktní
- Osmitaktní
- N-taktní

Další dělení je podle počtu současně buzených fází

- Magnetizace jedné fáze
- Magnetizace dvou fází

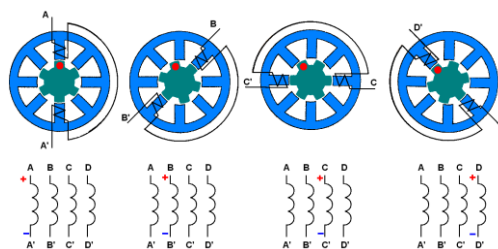
### 2.3.1 Unipolární řízení krokových motorků

U krokových motorů s pasívním rotorem nemá smysl měnit polaritu magnetického pole jednotlivých fází. Rotor je z magneticky měkkého materiálu. Většinou je složený z plechů aby se zmenšily ztráty v magnetickém obvodu.



Obr. 10 Základní princip řízení motoru s pasívním rotorem [9]

Výkonová část řídicí elektroniky je tím pádem jednodušší. Buzení jednotlivých fází probíhá v předem stanoveném pořadí. K jednotlivým fázím se postupně připojí stejnosměrné napětí. Otáčení rotoru výrazně pomáhá rozdílný počet pólů rotoru a statoru. Díky tomuto rozdílnému počtu mají „neaktivní“ póly rotoru různou vzdálenost k statorovým pólům.



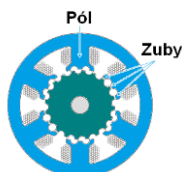
Obr. 11 Čtyřtákní unipolární řízení krokového motoru s pasívním rotorem s magnetizací jedné fáze

Pro správný chod krokového motoru s pasívním rotorem je důležité, aby byla v příštím kroku zaktivována dvojice statorových pólů, které mají menší vzdálenost k dvojici rotorových pólů. Rotor se natáčí tak, aby mezi aktivními cívkami procházel co největší magnetický tok. Celý princip je zachycen na obrázku **Obr. 11**. Zobrazený typ krokového motorku má velikost kroku danou podle následujícího vzorce.

$$\Delta\Phi = \frac{2\pi}{m \cdot N} \quad (1)$$

$\Delta\Phi$  ... velikost kroku  
 $m$  ... počet fází vinutí  
 $N$  ... počet zubů rotoru

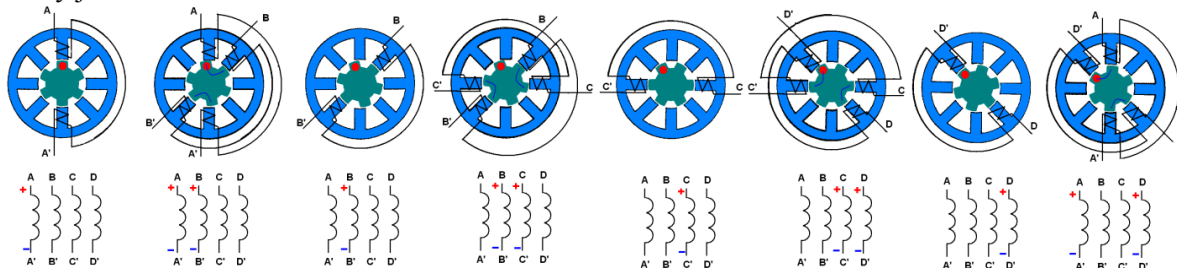
Při dosazení počtu fází rovno čtyřem a počtu pólů rovno osm vyjde úhel kroku  $15^\circ$ . Tento motor má 24 kroků na otáčku. Aby se zvýšil počet kroků u tohoto typu krokového motorku, přidávají se na pólové nástavce další zuby.



Obr. 12 Princip zvýšení počtu kroků reluktančního krokového motoru zvýšením počtu zubů na pólech

Pro zvýšení počtu kroků se také využívá více statorové motory, kdy motor složený z více statorů uložených za sebou. Tyto statory jsou navzájem pootočené a tím se získá lepší rozlišení. Tento princip je využit u hybridních motorů, které jsou popsány v odstavci 2.2.3.

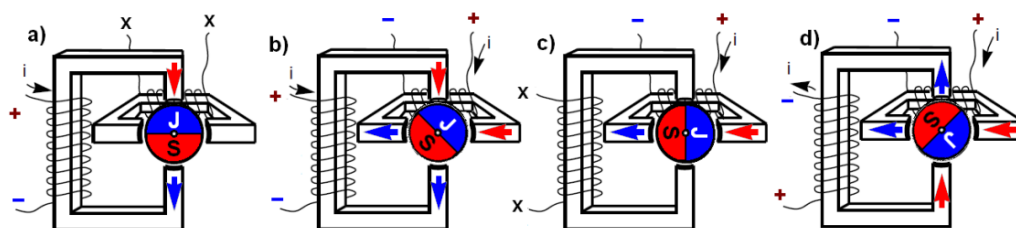
Zvýšení počtu kroků je možné dosáhnout také elektronickou cestou. Například současná magnetizace dvou fází. Krok se potom zdvojnásobí. Je to však vykoupeno složitějším řízením přepínání jednotlivých fází. Tak jako v předchozím případě, kdy bylo krokováno po celém kroku, je velmi důležité dodržet správný sled přepínání jednotlivých fází. Jak je vidět na obrázku, rotor se při tomto chodu pohybuje po menších krocích. Důvodů proč využít rozdělení základního kroku je víc. Mimo dosažení vyššího počtu kroků se tímto způsobem dosáhne i určitého zklidnění chodu motoru a sníží se vliv vnitřních rezonancí. Jo to tím, že zvlnění momentové charakteristiky mezi jednotlivými kroky je menší.



Obr. 13 Osmitákní unipolární řízení krokového motoru s pasívním rotorem s magnetizací dvou fází

### 2.3.2 Bipolární řízení krokových motorků

Pro krokové motorky, které mají rotor tvořený permanentním magnetem, se používá bipolární řízení. Díky permanentnímu magnetu je kolem rotoru magnetické pole. Změnou polaritu napájení cívek statoru se mění polarita statorového magnetického pole. Základní princip je znázorněn na následujícím obrázku. Krokování po celém kroku je na obrázku a) a obrázku c). Vložený půlkrok, kdy se budí dvě cívky najednou, je potom znázorněn na obrázku b) a d).



Obr. 14 Bipolární řízení krokových motorků s aktivním rotorem [10]

Skutečné krokové motory se na první pohled mohou lišit počtem vyvedených vodičů. Je to dáno tím, že jsou u některých motorů vyvedeny pouze konce jednotlivých vinutí, u některých se vyvádí i středy těchto cívek. Některé motory mají cívky rozděleny úplně.

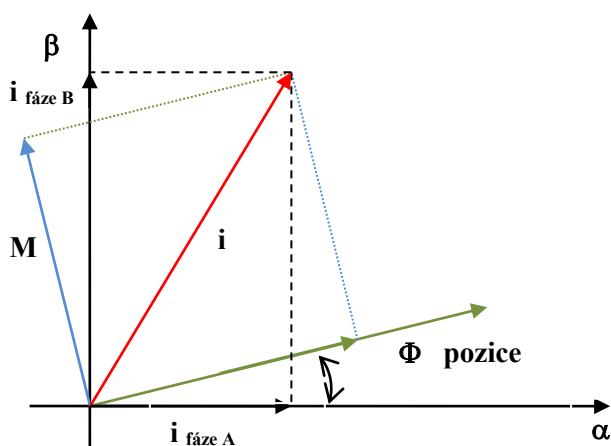
4-vodičové bipolární zapojení	6-vodičové unipolární zapojení	6-vodičové bipolární sériové zapojení
8-vodičové unipolární zapojení	8-vodičové bipolární sériové zapojení	8-vodičové bipolární paralelní zapojení

Obr. 15 Různé druhy zapojení krokového motorku [11]

### 2.3.3 Mikrokrokování

Výrobci krokových motorků se snaží stále zlepšovat vlastnosti svých výrobků. Zvyšuje se moment, přesnost, ale i rychlost. Zvýšení počtu základních kroků je již obtížnější. Počet nástavců, zubů, ale i fází je limitován obtížností výroby. S počtem základních kroků roste i cena motorku.

Moderní aplikace však vyžadují stále vyšší přesnost a tím pádem i rozlišení. Pokud se zvýší moment motorku, je možné vynechat převodovku do pomala, která přidává další chybu do přesného pohybu. Zvláště při změně směru. Při vynechání této převodovky je však problém se zachováním základní přesnosti.



Obr. 16 Vektorový diagram pootočení

Aby bylo možné zvýšit počet kroků na jednu otáčku, používá se různých metod, které dokáží základní mechanický krok dále rozdělit na více úseků. Jedná se o různé spínání základních fází jako například čtyřtaktní, osmitaktní spínání a tak dále. Dalším způsobem, který dokáže velmi zásadně změnit počet kroků na otáčku je takzvané mikrokrokování. Tímto způsobem je možné rozdělit jeden základní krok na 16, 32, 64 až 128 kroků.

Princip se zakládá na rozdělení poměru budících proudů mezi dvě sousední fáze. Pro dvoufázový hybridní motor je pro jeden plný krok nutné změnit magnetické pole o  $90^\circ$ . Pokud se má moment motoru měnit sinusově s polohou rotoru a mezní vazební moment má být úměrný proudu fází, je možné postupovat podle následujících rovnic. [12]

$$M_A = -k_T \cdot i_A \cdot \sin(N \cdot \Delta\Phi) \quad (2)$$

$$M_B = -k_T \cdot i_B \cdot \sin\left(N \cdot \Delta\Phi - \frac{\pi}{2}\right) \quad (3)$$

$\Delta\Phi \dots$  velikost kroku

$k_T \dots$  konstrukční konstanta motorku

$i_A, i_B \dots$  proudy jednotlivých fází

Při plném kroku se výsledný vektor magnetického pole pootočí o  $\varphi = 90^\circ$  elektrických. V režimu mikrokrokování se tento úhel rozdělí na  $n$  mikrokroků. Velikost proudu potom musí odpovídat velikosti úhlu podle následující rovnice.

$$i_A = I \cdot \cos\left(\frac{k \cdot \varphi}{n}\right) \quad (4)$$

$$i_B = I \cdot \cos\left(\frac{k \cdot \varphi}{n} - \frac{\pi}{2}\right) \quad (5)$$

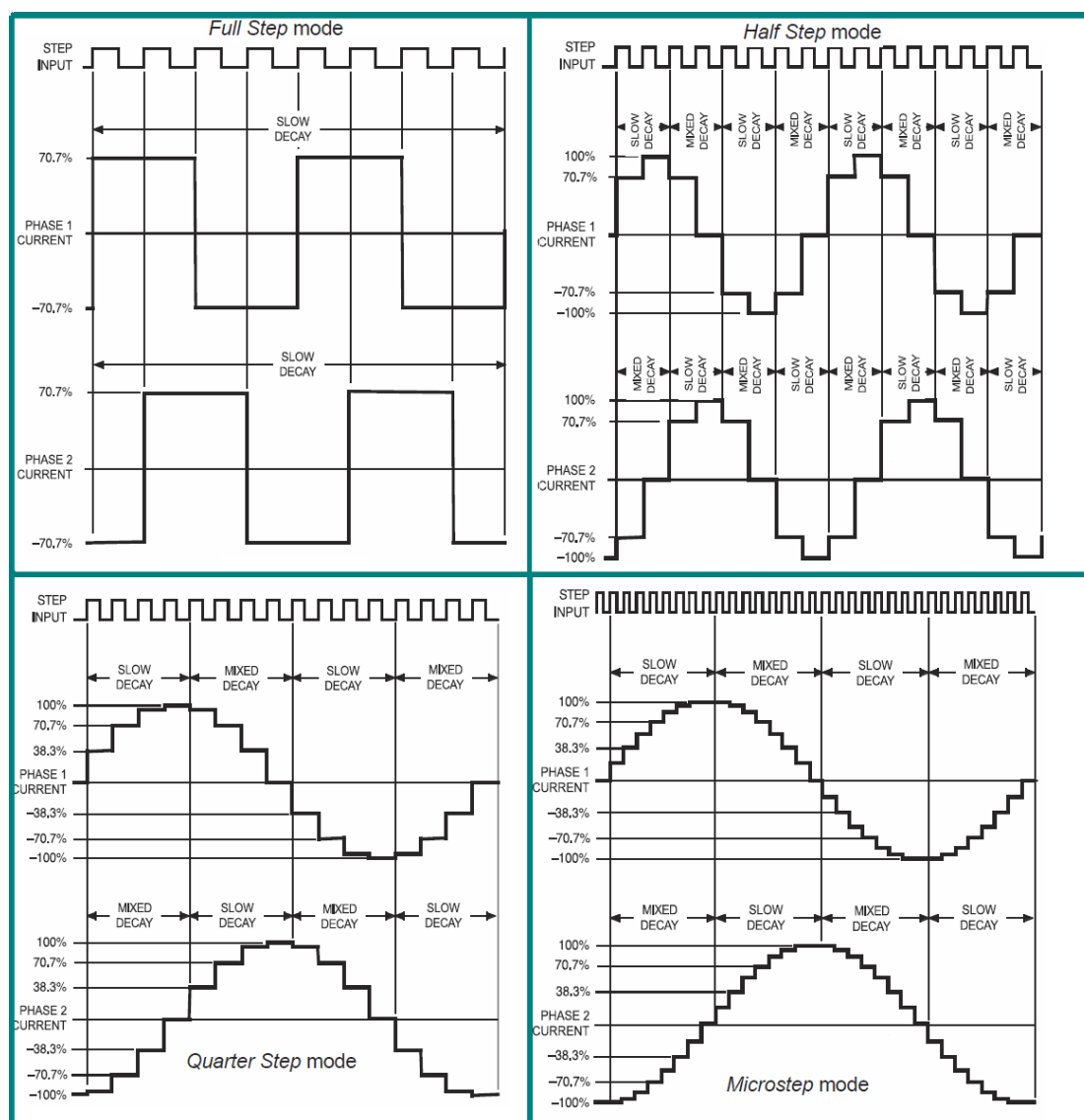
kde  $k = 1, 2, 3 \dots$

Při tomto řízení je možné dosáhnout konstantního vazebního momentu, který odpovídá momentu, který je získán při buzení jedné fáze. Navíc je zde výhoda v tom, že se zmenší zvlnění momentu mezi jednotlivými kroky, poklesne hluk i přirozená rezonance. Dále se zvýší účinnost motoru, protože se zlepší využití momentu. [12]

Nevýhodou mikrokrokování je vyšší náročnost na řízení motoru. Je nutné měnit velmi rychle poměr proudů ze sousedních dvou fází. To se děje buď výpočtem, nebo pomocí předem vypracovaných tabulek. V současné době pokročila výkonnost malých procesorů na takovou míru, že toto řízení přestalo být složitým problémem. Většina výrobců řídicích jednotek pro krokové motory již tuto funkci nabízí. Liší se maximální počet mikrokroků.

Mikrokrokování je způsob jak dosáhnout velmi jemného a přesného pohybu. Pro úplnost je nutné dodat, že přesnost se nezvyšuje, protože záleží na přesnosti základního kroku, který je pouze rozdělen na více kroků.

Různé způsoby pro zvýšení počtu základních kroků dvoufázového hybridního motoru jsou zobrazeny na následujícím obrázku. Z grafů je vidět, že při použití mikrokrokování jsou omezeny ostré hrany a průběh se blíží sinusovce. Tím je krokový motor ušetřen velkých rázů a pracuje tiše a plynule. V některých aplikacích se využívá mikrokrokování právě z těchto důvodů.



Obr. 17 Různé metody pro zvýšení počtu kroků dvoufázového hybridního krokového motoru [13]



### 2.3.4 Statická momentová charakteristika

Jelikož se využívá u krokového motoru stav, kdy motor stojí a drží svoji polohu, uvádí se pro tento motor mimo dynamické momentové charakteristiky také statická momentová charakteristika.

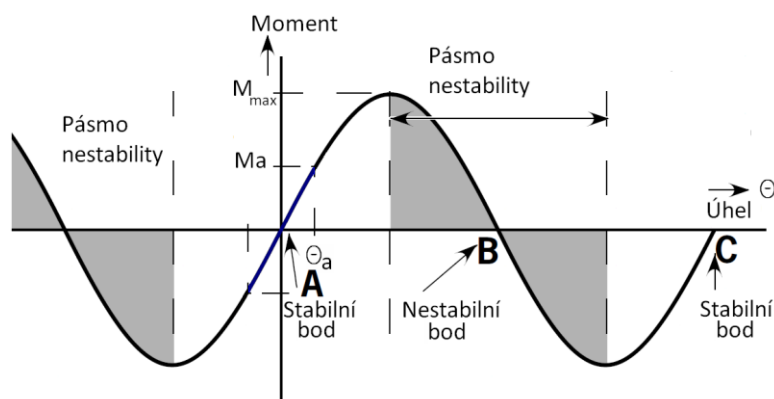
Na následujícím obrázku je znázorněna právě tato statická momentová charakteristika. Motor působí proti vnější síle, která se snaží natočit rotor o úhel  $\Theta$ . Moment v závislosti na úhlu natočení stoupá sinusovým průběhem až k maximálnímu momentu. Při překročení tohoto úhlu nastává pásmo nestability a moment stejným průběhem klesá. V polovině úhlu kroku je moment nulový. Nachází se zde takzvaný nestabilní bod. Za tímto bodem začne být rotor přitahován dalším pólem. Pokud motor stojí, působí na rotor nějaká síla. Ta je vyrovnávána momentovou silou. Tento klidový moment však znamená posun oproti bodu A o úhel  $\Theta_a$ . Je to statická chyba.

Pro upřesnění je nutno dodat, že ve skutečnosti není průběh statické chyby přesně sinusový, ale odchylka od sinusovky je pro běžné výpočty zanedbatelná. Pokud si pomůžeme sinusovkou, je pak výpočet úhlu vychýlení vcelku jednoduchý.

$$M = -M_M \cdot \sin(p \cdot \theta) \quad (6)$$

$$\theta_A = \frac{\sin^{-1}\left(-\frac{M}{M_M}\right)}{p} \quad (7)$$

$p \dots$	Počet rotorových zubů
$M \dots$	Moment působící na rotor
$M_M$	Maximální moment



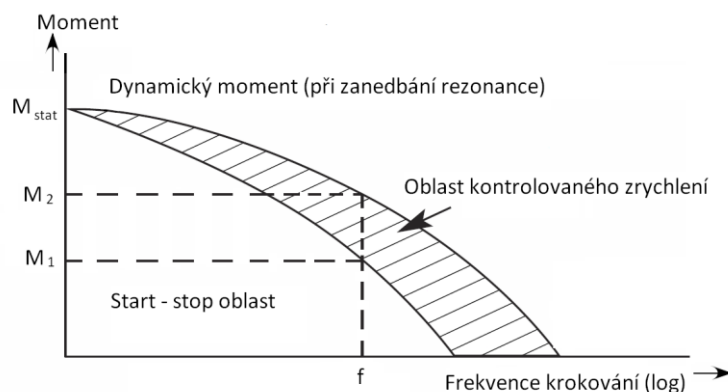
Obr. 18 Statická momentová charakteristika krokového motoru [14]

### 2.3.4 Dynamická momentová charakteristika

Dynamická momentová charakteristika má také svoje zvláštnosti oproti jiným typům elektromotorů. Krokový motor se využívá z důvodu přesného nastavení polohy bez nutnosti použít snímače polohy. Proto je problém pokud se během pohybu ztratí krok. Tato chyba může nastat jednak působením velkého vnějšího momentu, ale také při rychlém rozběhu vlivem setrvačných sil.

Aby nedocházelo k chybám při rozběhu, používá se takzvaný řízený rozběh, který umožní dosáhnout ve výsledku mnohem vyšší rychlosti motoru s nižší ztrátou momentu. V momentové charakteristice jsou proto dvě křivky. Bere se v úvahu průběh s řízeným rozběhem. V charakteristice je také vyznačen rozdíl momentů na stejné frekvenci. Nižší průběh maximálního momentu je výsledkem připojení pracovní frekvence hned po startu, druhý průběh nastává, pokud se frekvence plynule zvyšuje, až k frekvenci pracovní. Bylo by dobré zmínit i závislost momentu na velikosti napájecího napětí, které se projevuje při vyšších rychlostech. Je to dáno hlavně menší dobou pro nabití cívky dostatečným nábojem.

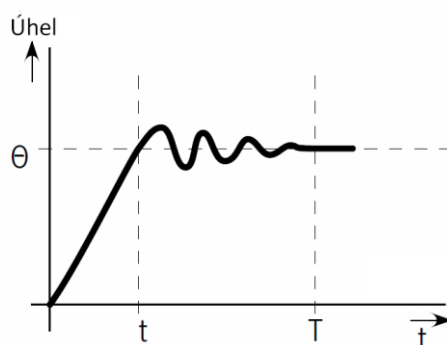




Obr. 19 Dynamická momentová charakteristika krokového motoru [14]

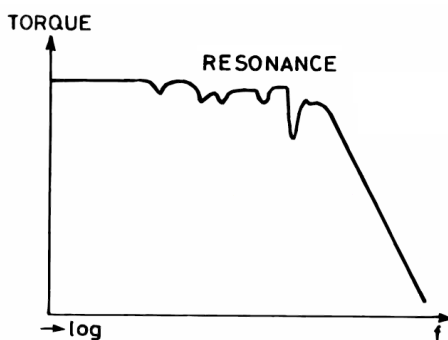
Pokud se připojuje pracovní frekvence hned po startu, pohybuje se výsledný moment pouze v takzvané start-stop oblasti.

U krokových motorů je skutečný průběh momentové charakteristiky odlišný než zobrazuje předchozí obrázek. Charakteristika bývá zvlněná a objevují se úzké oblasti, kdy je pokles charakteristiky značný. Obr. 21



Obr. 20 Odezva na jeden krok [14]

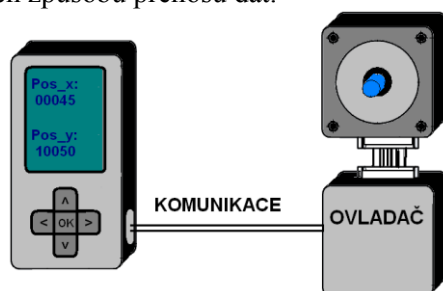
Důvodem je, že pohyb rotoru je víceméně složen z jednotlivých kroků. Není tedy úplně plynulý. Zvláště v malých rychlostech se projevuje kmitání rotoru kolem ustálené polohy při přechodu z jednoho kroku na druhý. V určitých oblastech momentové charakteristiky dochází k vnitřní rezonanci a moment klesá. K odstranění těchto jevů se používá rozdělení základního kroku, tlumiče na hřídeli, atd.



Obr. 21 Průběh momentové charakteristiky krokového motoru s poklesy vlivem rezonancí [15]

### 3. Komunikační protokoly

Jak již bylo uvedeno, krokové motory se uvádějí do pohybu různou posloupností přepínání jednotlivých cívek (fází). To zajišťuje elektronická řídicí jednotka zvaná též kontrolér nebo driver. Aby driver věděl, co má dělat, je dále řízena nadřazenou jednotkou nebo regulátorem. Tato nadřazená jednotka může potom ovládat více motorů najednou. Typickým příkladem je CNC stroj, který ovládá motory v osách x, y a z. Pro komunikace mezi nadřazenou jednotkou a jednotkou přepínající jednotlivá vinutí se používá několik různých způsobů přenosu dat.

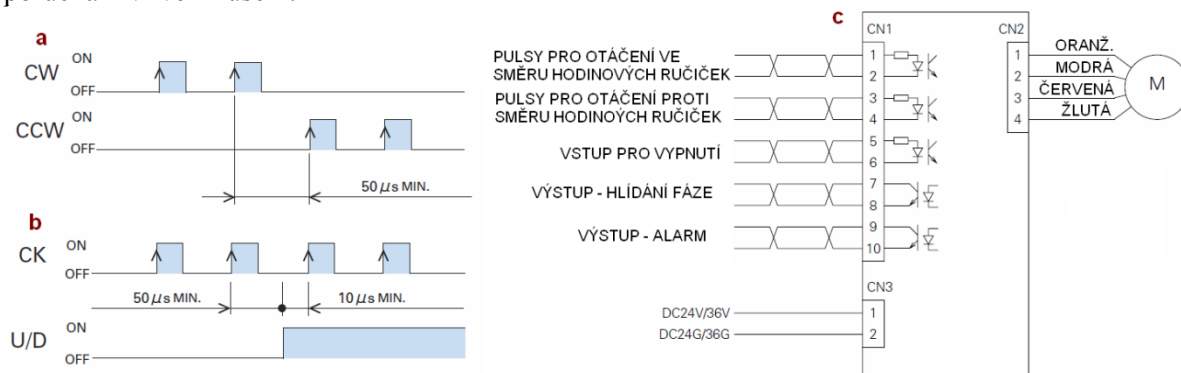


Obr. 22 Ovládání krokových motorků - nadřazená jednotka, ovladač, motor a komunikace mezi nimi

#### 3.1. Řízení pomocí pulsů, směru a vypnutí „disable“

Tato komunikace nebo spíše řízení je asi nejrozšířenější a nejjednodušší. Ovládání pobíhá pomocí několika vodičů. Každý z vodičů přenáší jiný signál. První přenáší pulsy, které odpovídají jednotlivým krokům. Pokud nejsou přítomny pulsy, motor stojí. Další signál přenášený po samostatném drátě je signál směru. Podle toho v jaké je napěťové úrovni (vysoké nebo nízké) je dán směr pohybu (dopředu nebo vzad).

Poslední signál „disable“ vypíná driver nebo spíše proud do cívek motoru. Motor pak přestává elektronicky (magneticky) držet svoji polohu. Frekvence pulsů bývá u dnes běžně používaných driverů maximálně kolem 40kHz. V praxi se používá frekvencí nižších podle použitých motorů. Tento způsob řízení bývá různě modifikován tak, aby nedocházelo ke ztrátě kroků nebo jiným poruchám vlivem rušení.



Obr. 23 Příklad ovládání ovladače pro krokový motor pomocí pulsů

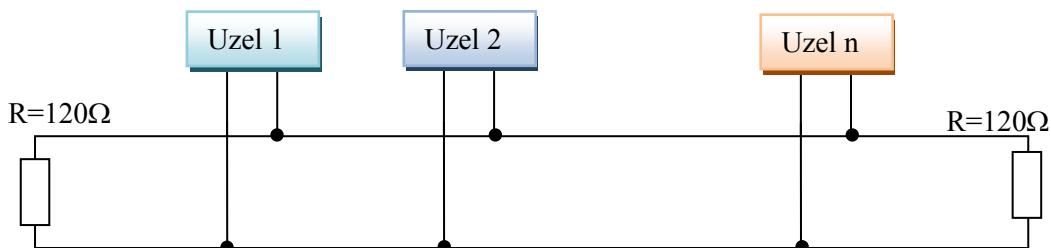
a) samostatné pulsy pro kroky dopředu a dozadu

b) Společné pulsy a signál ovládající směr

c) Driver pro variantu (a) [7]

### 3.2. Řízení pomocí sběrnice CAN BUS

Pro komunikaci ve vozidlech byla během osmdesátých let minulého století vyvinuta firmou Bosch zvláštní sériová komunikace zvaná CAN (Controller Area Network). V této době začaly pronikat do automobilového průmyslu různé palubní počítače a vznikl požadavek na spolehlivou komunikaci s čidly. Samozřejmě vzniklo pro tento účel více alternativ a více druhů komunikací, ale systém CAN se postupně prosadil. První skutečné použití bylo v roce 1992 ve vozidlech Mercedes-Benz. O rok později byla zveřejněna norma ISO11898, která tuto komunikační síť popisuje. Sběrnice CAN je velmi odolná proti rušení a používá se v současné době nejen v automobilech, lodích nebo lokomotivách. Pro svoji odolnost a spolehlivost se začala používat i v průmyslových aplikacích. Během několika let se stala velmi oblíbenou a vedle RS485 je dominující sběrnice pro přenos dat v tomto odvětví.



Obr. 24 Příklad zapojení komunikace CAN

Pro fyzickou vrstvu CAN byly vyvinuty a jsou vyráběny integrované odvoody – drivery, které jsou připojeny na dvoudrátovou sběrnici o charakteristické impedanci  $120\Omega$ . Maximální délka sběrnice se udává kolem sta metrů, ale pro vyšší rychlosti je to kolem čtyřiceti metrů. Některé zdroje udávají i větší délky, ale při těchto délkách rychlost i spolehlivost komunikace klesá.

Rychlosti přenosu používané u vysokorychlostních linek (podle ISO11898-2) jsou max. 1 Mbit/s. U linek odolných proti chybám (podle ISO11898-3) je to maximálně 125kbit/s. Norma udává i počet uzlů. Používá se maximálně 64 uzlů na jeden segment.

Pro bitové kódování se používá NZR (Non Return to Zero). Protokolu CAN definuje dvě hodnoty bitů na sběrnici - dominant a recessive. Je to ekvivalent logických úrovní, jejichž hodnoty nejsou určeny a skutečná úroveň záleží na konkrétní realizaci fyzické vrstvy.

Protože se jedná o symetrické vedení, jsou tyto dvě úrovně udávány jako velikost rozdílového napětí. Pro recessive velikost rozdílového napětí  $V_{\text{diff}} = 0 \text{ V}$  a pro úroveň dominant  $V_{\text{diff}} = 2 \text{ V}$ .

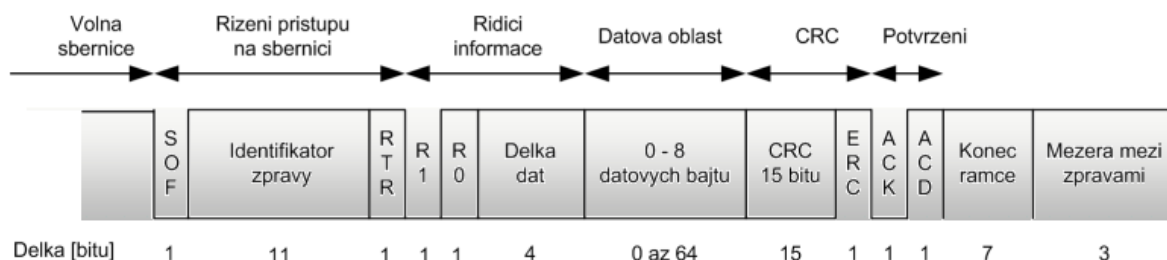
Vzhledem k tomu, že se jedná o síť typu multimaster, každý z účastníků může zahájit vysílání, jakmile je připraven a síť je v klidovém stavu (bus free). Kdo přijde první, ten vysílá. Ostatní mohou vysílat až poté, co je zpráva odvysílána. Vyjimku tvoří chybové rámce, které se dají vysílat okamžitě po identifikaci chyby kterýmukoli účastníkem.

Zahájí-li vysílání současně několik uzlů, pak přístup na sběrnici získá ten, který přenáší zprávu s vyšší prioritou (nižším identifikátorem). Identifikátor je uveden na začátku zprávy. Každý vysílač porovnává hodnotu právě vysílaného bitu s hodnotou na sběrnici a zjistí-li, že na sběrnici je jiná hodnota než vysílá (jedinou možností je, že vysílač vysílá recessive bit a na sběrnici je úroveň dominant), okamžitě přeruší další vysílání. Tím je zajištěno, že zpráva s vyšší prioritou bude odeslána přednostně a že nedojde k jejímu poškození, což by mělo za následek opakování zprávy a zbytečné prodloužení doby potřebné k přenosu zprávy. Uzel, který nezískal při kolizi přístup na sběrnici, musí vyčkat, až bude sběrnice opět ve stavu Bus free, a pak zprávu vyslat znovu. [16]

Protokol CAN se vyznačuje velmi silným zabezpečením přenášených dat. Používají se následující procedury:

- monitoring - kontrola vyslaného bitu se stavem na sběrnici
- CRC kód – kontrolní součet pomocí Cyclic Redundancy Check
- vkládání bitu – při pěti po sobě jdoucích stejných bitech se vkládá jeden bit opačný
- kontrola zprávy – má-li některý bit nepovolenou hodnotu, je generována chyba
- potvrzení přijaté zprávy - každé zařízení, připojené ke sběrnici musí správně přijatou zprávu potvrdit. Činí tak změnou bitu v poli ACK (1 bit) z recessive - vysílané vysílačem na dominant [B10]

CAN je sériový komunikační protokol umožňující vzdálené řízení systémů v reálném čase s vysokou mírou zabezpečení proti chybám. Jedná se o protokol typu multi-master, kde každý uzel sběrnice může být master a řídit tak chování jiných uzlů. Tato vlastnost přináší zjednodušení řízení a zvyšuje spolehlivost. Při poruše jednoho uzlu může zbytek sítě pracovat dál.



Obr. 25 Přenášená zpráva po sběrnici protokolem CAN 2.0A

Pro řízení přístupu k médiu je použita sběrnice s náhodným přístupem, která řeší kolize na základě prioritního rozhodování. Po sběrnici probíhá komunikace mezi dvěma uzly pomocí zpráv (datová zpráva a žádost o data), a management sítě (signalizace chyb, pozastavení komunikace) je zajištěn pomocí dvou speciálních zpráv (chybové zprávy a zprávy o přetížení).

Tab. 2 Popis zprávy sběrnice CAN [16]

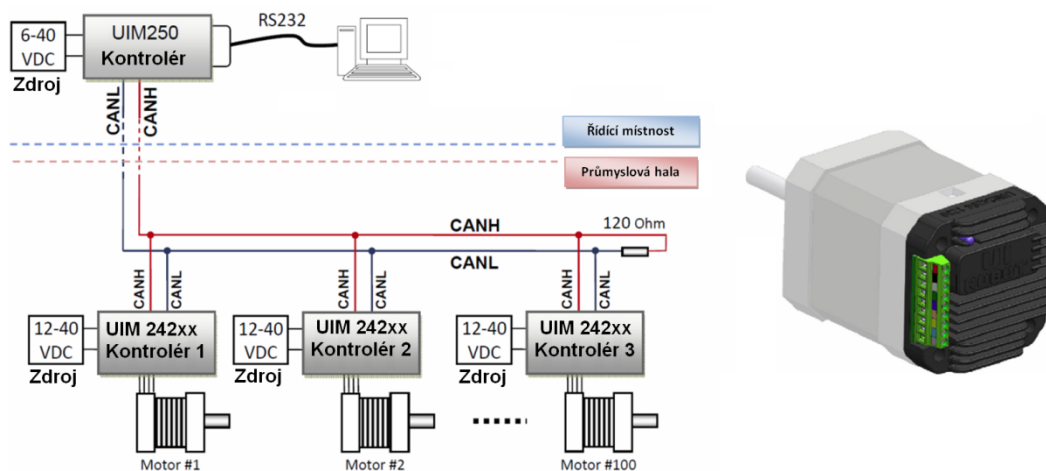
	Délka (b)	Název	Popis
1	1	SOF Start of Frame	Začátek zprávy
2	11	Arbitration Field	Řízení přístupu ke sběrnici a identifikátor zprávy, určuje prioritu zprávy a význam přenášené zprávy
3	1	RTR Remote Request	Slouží k rozlišení zprávy zda jde o datovou zprávu (dominant) nebo žádost o přístup ke sběrnici (recessive)
4	1	Control Field R0	Rezervováno
5	1	Control Field R1	Rezervováno
6	4	Data Len	Délka datové zprávy
7	1-64	Data Field	Datová oblast
8	15	CRC	Zabezpečovací kód
9	1	ERC	CRC oddělovač (dominant)
10	1	ACK	Potvrzení
11	1	ACD	Potvrzení
12	7	End of Frame	Konec zprávy (recessive)
13	3	Interframe Space	Mezera mezi zprávami (recessive)

Zprávy vysílané po sběrnici protokolem CAN neobsahují žádnou informaci o cílovém uzlu, kterému jsou určeny, a jsou přijímány všemi ostatními uzly připojenými ke sběrnici. Každá zpráva je označena identifikátorem, který udává význam přenášené zprávy a její prioritu. Protokol CAN zajišťuje, aby zpráva s vyšší prioritou byla v případě kolize dvou zpráv doručena. Identifikátor slouží také k tomu, aby uzel přijímal pouze ty zprávy, které se ho týkají. [16]

Specifikace protokolu CAN definuje čtyři typy zpráv:

- datová zpráva
- žádost o data
- zpráva o chybě
- zpráva o přetížení

Jak již bylo zmíněno, sběrnice CAN je vyvinutá pro komunikaci v prostředí se silným elektromagnetickým rušením. Proto se přímo nabízí využití pro řízení krokových motorků. Sběrnice CAN se často využívá i v prostředí, kde by měla RS485 problémy.



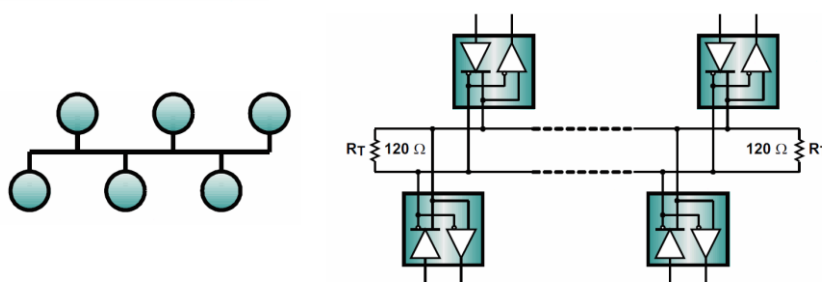
Obr. 26 Příklad ovladače krokových motorků se sběrnici CAN 2.0A

Pro použití v oblasti řízení krokových motorků nabízejí výrobci integrovaných obvodů drivery s CAN rozhraním. Například ON Semiconductor vyrábí pod označením AMIS-30523 driver s mikrokrokováním, který je pro tuto komunikaci připraven. Na trhu jsou také hotové ovladače, které je možné ovládat po sběrnici CAN.

### 3.3. Řízení pomocí sběrnice RS485

V roce 1983 vydalo sdružení EIA popis standardu komunikační sběrnice RS485. Tato symetrická poloduplexní komunikace využívá k přenosu dva dráty A a B. Je velmi odolná vůči rušení, a proto je používána hlavně pro průmyslové aplikace.

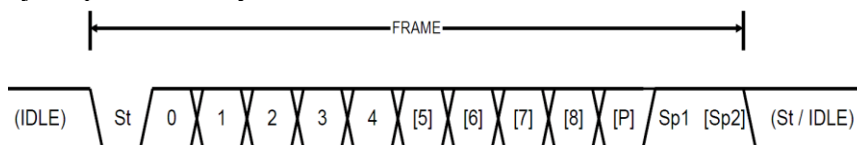
Maximální délka jedné větve je podle normy 1200m. Na jedné větvi může být maximálně 32 zařízení. Rychlost je dána délkou linky. Pro délku kolem 12m je možné použít rychlost až 10Mb/s, pro maximální délku je rychlost omezena na 100kb/s. Topologie sítě je podélná z jednoho zařízení do druhého a na obou koncích vedení by měly být zakončovací odpory  $120\Omega$  pro impedanční přizpůsobení linky. K přenosu se používá kroucený pár.



Obr. 27 Topologie a příklad zapojení sítě RS485 [17]

Aby bylo možné použít jiný typ topologie, například hvězdu, používá se takzvaných opakovačů (repeater). Pomocí opakovačů je možné zvyšovat přenášet data na delší vzdálenosti. K oddělení jednotlivých segmentů sítě se také používají směrovače (routery). Ale to je už spíš otázkou nadřazeného protokolu, zda má síťovou vrstvu, která tyto routery obsluhuje (například protokol BACnet – MSTP).

Přenos dat se uskutečňuje pomocí 7 nebo 8 bitových rámců (telegramů) se startbitem, jedním nebo více stopbity. Může být aktivován i paritní bit. Startbit je logická nula, stopbit a neaktivní stav je reprezentován logickou jedničkou. Komunikace je asynchronní, a proto musí být přijímač i vysílač nastaveny na stejnou přenosovou rychlost.



Obr. 28 Rámec přenosu jednoho byte informace [18]

St	Start bit, vždy v nule.
(n)	Datové bity 7 nebo 8.
P	Paritní bit. Může být lichá nebo sudá parita.
Sp	Stop bit, vždy v jedničce

Na symetrické lince je potom stav jedničky a nuly reprezentován různou polaritou rozdílového napětí mezi vodiči. Díky nízké impedanci teče linkou dostatečný proud na to, aby byl přenos odolný cizímu rušení.



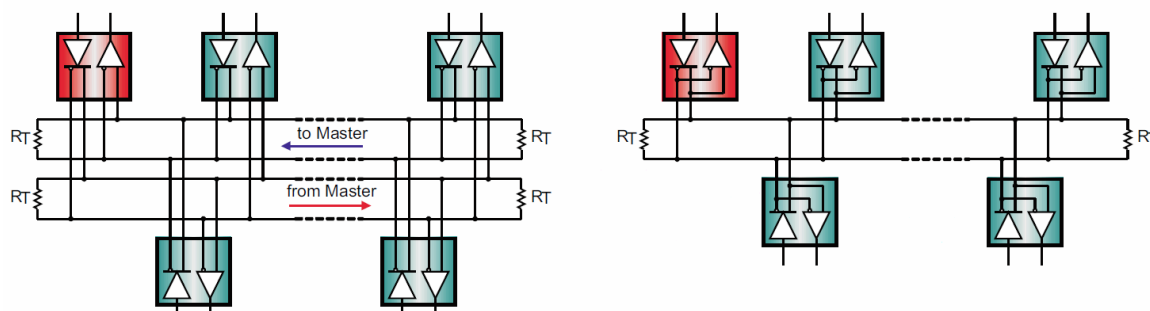
Obr. 29 Minimální úrovně pro sběrnici RS485 [17]

Jelikož je RS-485 poloduplexní komunikace, musí být zaručeno, aby v jednom okamžiku vysílal pouze jeden master. To zajišťuje komunikační protokol. Komunikační protokoly používané na lince 485 jsou například MODBUS RTU, MODBUS ASCII, BACnet, MSTP, PROFIBUS DP/FMS a tak dále.

Tyto protokoly fungují tak, že je na lince pouze jeden master, který se musí dostatečně často ptát všech ostatních zařízení (slaves), které jsou rozlišeny různou adresou. Ostatní zařízení potom odpovídají v přesně vymezeném časovém úseku tak, aby se nedostali do kolize s masterem, nebo jiným zařízením, které také odpovídá.

Dalším způsobem je předávání tokenu (slova), kdy jeden master předá slovo jinému zařízení, ze kterého se na určitou dobu stane také master.

Existuje i verze full-duplex, kdy je možné komunikovat najednou v obou směrech. Tato komunikace používá standard RS-422. Problémem je větší počet drátů. V praxi se však často ukazuje, že každý drát navíc může být při realizaci velký problém. Nejen co do investice, ale hlavně při finálním zapojování aplikace, kdy je často nutné správně pospojovat správně velká množství vodičů. Navíc se u této komunikace používají párované vodiče, a právě špatné párování je velmi častou chybou u takovýchto aplikací.



Obr. 30 Full-duplex sběrnice RS422 a Half-duplex sběrnice RS485 (vpravo) [17]

### 3.4. Použití sběrnice RS485 pro řízení krokových motorků

Jak již bylo zmíněno, sběrnice RS-485 je odolná průmyslová sběrnice, proto se nabízí její použití pro ovládání krokových motorů. Právě motory obecně se v poslední době stávají velmi často zdrojem rušení. Důvodem je použití pulzních modulací pro regulaci proudu, otáček, momentu nebo napětí. Velký pokrok ve vývoji a výrobě polovodičů pomohl k snadnější realizaci pulsně-šířkové modulace. To je velmi významné pro snížení ztrát a pro využití různých typů motoru tam, kde by tyto motory neměly dříve uplatnění.

Pro účinnou PWM modulaci je snaha vyrobit co nejostřejší hrany při spínání jednotlivých tranzistorů tak, aby byly tyto tranzistory co nejvíce chráněny před ohříváním vlivem ztrátových výkonů, na druhou stranu se objevuje problém s velkým nárůstem rušení právě díky ostrým hranám. Odstranění rušení vyžaduje použití filtrů (sinusových,  $du/dt$  atd.). Tyto filtry však navyšují cenu a vcelku znatelně zvyšují ztráty. Proto bývá výsledkem nějaký kompromis.

V současné době je to tak, že se v průmyslovém prostředí zvyšuje počet zdrojů rušení. Současně se zvyšuje využití různých komunikačních protokolů. Proto je kladen důraz na to, aby byl průmyslový komunikační protokol co nejodolnější. Je výhodnější používat například inteligentní čidla, která se připojují přímo do sítě RS485, než čidla, která se připojují dlouhým kabelem do jednotky. Vedení se musí kompenzovat (například vést zvlášť napájení a snímané napětí), musí se odfiltrovat rušení, čímž se sníží přesnost a rychlost. Pokud se převede měřená hodnota do digitální hodnoty co nejbližší čidlu, případné poruchy na komunikaci je možné zjistit a zprávy opakovat. Navíc stačí jedno vedení na více čidel.

Tento trend je patrný i u ovládání. Je nutné dovést požadovanou hodnotu ve formě čísla co neblíže motoru, nebo jiného akčního zařízení a až tam ho převést na nějaký regulační zásah, nebo změnu veličiny. Ovládání pomocí analogových signálů (např. 0-10V nebo 4-20mA) pomalu ustupuje.

Využití komunikační sběrnice RS-485 proto velmi narůstá. Není se proto čemu divit, že i výrobci driverů pro krokové motory začleňují do svých výrobků nějaký typ „inteligentního“ ovládání.

V následujícím textu jsou popsány dva typy driverů, které pro ovládání využívají komunikační protokol RS485.

#### 3.4.1. Kontroléry s procesorem M1486

Firma MICROCON je na českém trhu známá jako dodavatel krokových motorků a ovladačů. Tato firma nabízí ovladač pro krokový motor, který je možné ovládat pomocí ASCII příkazů. Tyto příkazy jsou do kontroléru posílány pomocí asynchronního sériového portu rychlostí 300 až 9600 Bd. Tím pádem je možné připojit celý ovladač na sběrnici RS485 a ovládat ho na větší vzdálenost.

Několik příkazů, které používá tento kontrolér je pro představu uvedeno v následující tabulce. Kompletní sada a popis je na stránkách firmy MICROCON.

#### Základní charakteristika

- uživatelsky přátelský výkonný soubor více než 50 povelů
- provádění povelů OFF-line
- programovatelná maximální rychlost, START/STOP rychlost a zrychlení
- plynulé zrychlení a dojezd
- rychlost do 40 000 kroků/sec
- délka dráhy až 16 milionů kroků
- sériový vstup, až 16 procesorů může být připojeno na jeden port
- univerzální logické vstupy a výstupy
- po resetu obvodu výkonový modul setrvává v předchozí pozici



Tab. 3 Příklad použitých příkazů kontroléru M1486 [19]

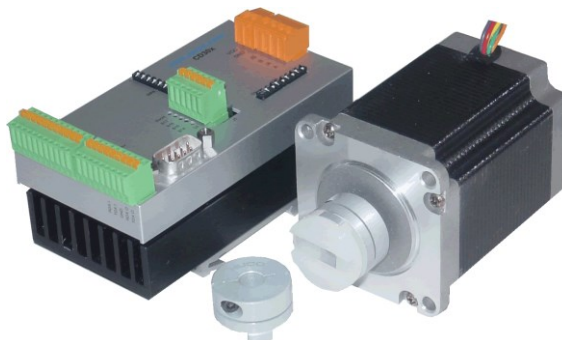
Povel	Popis
<b>\</b>	"Reset" uvedení kontroléru do výchozího stavu
<b>@ (Num)</b>	"Address" zadané číslo návěstí se přiřadí této programové řádce
<b>A (Num)</b>	"Acceleration" zrychlení, rozsah = 1 až 65 000 kroků/s <sup>2</sup>
<b>B (Num)</b>	"Backward" zpět, zadání dráhy v negativním směru, rozsah = 1 až 16 000000
<b>C (Num 1 to 21)</b>	"Clear" nastav zadaný výstup do hodnoty logická nula, rozsah 1 až 21
.....	
<b>L(Num)</b>	"Loop" smyčka, opakuj provádění následujících instrukcí
<b>M (Num)</b>	"Microstepping" počet mikrokroků na celokrok v dolním pásmu rychlostí (až do 64)
<b>N (Num)</b>	"Number" výběr tvaru průběhu proudu při mikrokrokování
.....	
<b>X (Num)</b>	"indeX" volba kontroléru
<b>Z (Num)</b>	"Zero" čekej, dokud zadaný vstup nebude mít hodnotu logická nula
.....	
<b>' (Num)</b>	"Subroutine" podprogram
<b>. (Num)</b>	"End of subroutine" konec podprogramu

Pomocí těchto příkazů je možné řídit krokový motor velmi rychle, přesně a efektivně. M1486 umožňuje také sestavovat z jednotlivých příkazů programy. Například:

**S200** "Start/stop" počáteční rychlost 200  
**V1000** „Velocity“ rychlost 1000 kroků  
**A5000** „Acceleration“ zrychlení 5 000 kroků/s<sup>2</sup>  
**F25689** „Forward“ dráha dopředu 25 689 mikrokroků  
**L3** „Loop“ následující povely až po povel „End of loop“ budou provedeny třikrát  
**R** „Run“ Vykonej pohyb se zadanými parametry  
**W250** „Wait“ čekej 250 milisekund  
**E** „End of loop“ konec smyčky

Kontrolér typu M1486E je možné použít pro samostatný chod. A to tím způsobem, že program je dopředu uložen pomocí PC do paměti EEPROM kontroléru. Po zapnutí je následně program vykonáván autonomně bez nutnosti použití nadřazené jednotky.

Za zmínku stojí, že samotné řízení krokového motoru probíhá s ohledem na zachování maximálního momentu motoru. Driver dodržuje přednastavené zrychlení a je možné využít i mikrokrokování až do šestnácti mikrokroků na celokrok. [19]



Obr. 31 Programovatelná jednotka typu CD a krokový motor od firmy Microcon [19]



### 3.4.2. Kontrolér R356 od firmy RMS Technologies Inc

RMS Technologies Inc. se již několik desetiletí zabývá mimo jiné vývojem a výrobou driverů pro krokové motory. Za tu dobu dosáhla v této oblasti zajímavých výsledků. Její drivery mají v porovnání s konkurenčními výrobky nižší rušení a menší rozměry, při stejném proudu a momentu. Firma se také snaží vylepšit potlačení různých vnitřních rezonancí u krokových motorů.



Obr. 32 Kontrolér R356 od firmy RMS Technologies Inc [20]

Mezi jinými vyrábí tato firma driver (kontrolér), který je možné ovládat po lince RS485. Toto zařízení má označení R356. Jak je uváděno v dokumentaci, je to ovladač pro jednu osu. Lze k němu připojit dvoufázové krokové motorky. Pouzdro kontroléru je uzpůsobeno tak, aby bylo možné ho přišroubovat k zadnímu čelu krokového motoru a vytvořit tak kompaktní blok. Tím pádem není nutný speciální držák na tento kontrolér a navíc se propojení s motorem zkrátí na minimum.

Základní vlastnosti kontroléru R356:

- Pracovní napětí 12V až 40V
- Špičkový fázový proud 0,3A až 3A
- Maximální krokovací frekvence 2,147GHz
- Celokrok, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, 1/128, 1/256 rozdělení cellokroku
- Možnost samostatné činnosti bez připojení PC nebo nadřazené jednotky
- Programovatelné rychlosti
- Samostatně nastavitelné proudy pro zastavený motor a motor v pohybu
- Čtyři vstupy/výstupy a dva vstupy
- Vstupy napětí 0-5V DC (24V tolerant), vstupní proud až 700mA, pull-up 20kΩ
- Možnost adresování kontrolérů na lince RS485 (přepínatelná adresa 0-15)
- Volitelná přenosová rychlost 9600Bd, 19200Bd nebo 38400Bd
- Přenosové parametry 8bitů, bez paritního bitu, jeden stopbit

Kontrolér R356 má na svém vyvedeny tři konektory. Jeden čtyřpinový pro motor, jeden šestipinový pro encodér a jeden patnáctipinový pro konektor typu cannon, kterým se připojuje napájení, komunikace, vstupy a výstupy.



Obr. 33 Rozmístění konektorů na kontroléru R356 [20]

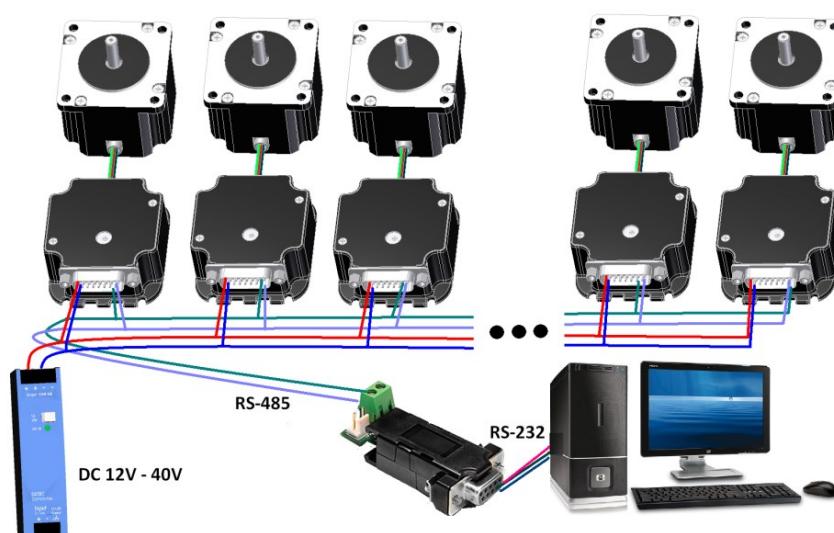
Z obrázku je patrné umístění jednotlivých konektorů. Na prostředním obrázku je dobře vidět otvor pro přepínač adresy zařízení. Na levém obrázku jsou popsány konektory pro motor a encoder. Na pravém obrázku je vidět umístění konektoru typu cannon. Jeho zapojení je popsáno v následující tabulce.

Tab. 4 Zapojení konektoru cannon 15 [20]

Pin	Barva	Popis
1	Zelená	Napájecí zem
2	Černá	1A On/Off výstup
3	Bílo-zelená	Přímý vstup
4	Žlutá	+5V DC Vstup pro optočleny – krokování a směr
5	Oranžová	Vstup 2 / Jog2 vstup
6	Žluto-bílá	Vnitřní napájení pro optosenzory
7	Oranžovo-bílá	Vstup (optočlen) 3/Opto1
8	Černo-bílá	RS485 A
9	Červená	Napájení - plusový pól +12V až +40V
10	Modrá	1A On/Off Výstup
11	Modro-bílá	Vstup krokování
12	Zeleno-bílá	Signálová zem
13	Bílá	Vstup 1 / Jog 1 vstup
14	Červeno-bílá	Vstup 4 (optočlen)/ Opto 1
15	Hnědá	RS485 B

Kontrolér je vybaven rychlým procesorem, který je možné ovládat pomocí ASCII příkazů. Tak jako u kontroléru od firmy MICROCON je možné sestavovat programy, které jsou uloženy v paměti EEPROM (resp. flash). Po zapnutí zařízení se tento program vykoná.

Kontroléry R356, které jsou připojeny na jedné lince a mají různou adresu, je možné ovládat skupinově jedním příkazem tak, aby se například dva motory rozběhly a ostatní zůstaly stát. Toto adresování bude podrobněji popsáno na další stránce.



Obr. 34 Zapojení několika driverů R356 v síti [20]

### 3.4.3. Ovládání kontroléru R356

Protože naše zařízení bude ovládat právě tento kontrolér, měl by se zde popsat základní princip tohoto ovládání. Není asi nutné sem kopírovat všechny příkazy tak, jak je popisuje výrobce. Celý seznam s podrobným vysvětlením uvádí výrobce na svých webových stránkách.

Zařízení je možné jednoduše ovládat pomocí PC například programem HyperTerminal. Postupně se složí řetězec z ASCII znaků, a ten se potom ukončí tlačítkem ENTER. Řetězce jsou skládány podle určitých pravidel DT protokolu.

/	0-15	Command strings	R	<CR>
---	------	-----------------	---	------

Tab. 5 Formát odesílaných dat do kontroléru [20]

Počáteční znak	Adresa	Příkaz	Běh (RUN)	Konec řetězce
/	0-15	Command strings	R	<CR>

Odesílaný příkazový řetězec je možné popsat následovně:

1. **Počáteční znak** jasně identifikuje začátek řetězce. Tento znak se dále v řetězci nevyskytuje.
2. **Adresa** se u driverů nastavuje otočným přepínačem, který je umístěn v otvoru pod vrchním krytem. Přepínačem je možné nastavit číslo 0 až 15. Při zadávání adresy do textového řetězce se postupuje tak, že čísla 1-9 jsou shodné, ale pro adresy 0 a 10 až 15, se využívají jiné znaky, než by se dalo očekávat. Seznam znaků je v následující tabulce.

Tab. 6 Přiřazení ASCII znaků k adresám [20]

Použité znaky při zadávání adresy																
Adresa	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Znak	@	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?

V praxi u větších aplikací je nutné posílat příkazy několika krokovým motorkům (resp. jejich driverům) najednou. Nastavit jim stejnou adresu není úplně nejlepší řešení, protože nedostaneme nikdy platnou odpověď. Zařízení R356 na to však pamatuje jednou vychytávkou. Má totiž speciální adresy pro více zařízení. Správným přiřazením adres je potom snadné ovládat například vícemotorové pojezdy CNC strojů. Hlavní výhodou je to, že oba, respektive čtyři motory dostávají příkaz současně, a proto nedochází ke zbytečným prodávám a ztrátě souběhu motorů.

Tab. 7 Přiřazení ASCII znaků pro společné ovládání více zařízení najednou [20]

Použité znaky pro ovládání dvou a více adres najednou													
Znak	A	C	E	G	I	K	M	O	Q	U	Y	J	Všechny
Adresa 1	1	3	5	7	9	11	13	15	1	5	9	13	
Adresa 2	2	4	6	8	10	12	14	16	2	6	10	14	
Adresa 3									3	7	11	15	
Adresa 4									4	8	12	16	

3. **Příkazy**, kterými je možné ovládat R356, je možné rozdělit do několika skupin. Podrobný popis těchto příkazů je obsáhlý a je dostupný na stránkách výrobce i s podrobným vysvětlením jednotlivých funkcí.

Tab. 8 Rozdělení příkazů [20]

Rozdělení příkazů do jednotlivých skupin		
Název skupiny	Popis	Použité znaky
HOMING & POSITIONING	Pozice a navádění	Z,z,A,f,P,D,T,F
VELOCITY & ACCELERATION	Rozběh a rychlost	v,V,c,L
SETTING CURRENT	Nastavení proudu	m,l,h
LOOPING & BRANCHING (Cont.)	Větvení programu a smyčka	r,G,M,H,S,N,n
PROGRAM STORAGE & RECALL	Uložení a volání programu	s,e
PROGRAM EXECUTION	Vykonání programu	R,X
MICROSTEPPING	Nastavení rozdělení kroku	j, o
ON/OFF DRIVERS	Zapnutí driveru	J
QUERY COMMANDS	Vyčtení různých dat	?,&,Q,T,max
BAUD CONTROL	Nastavení přenosové rychlosti	b

4. **Run a <CR>** ukončují ASCII řetězec. Poslední znak se v běžných komunikačních programech nezobrazuje. Jeho hodnota je 13 nebo 0x0D HEX.

Pokud přijme kontrolér zprávu, odešle odpověď. Tím potvrdí, že zprávě rozuměl. Zároveň může svojí odpovědí upozornit na případnou chybu, ke které při zadávání příkazu mohlo dojít. Odpovědi na příkazy jsou popsány v Tab. 9.

Tab. 9 Odpověď na příkaz [20]

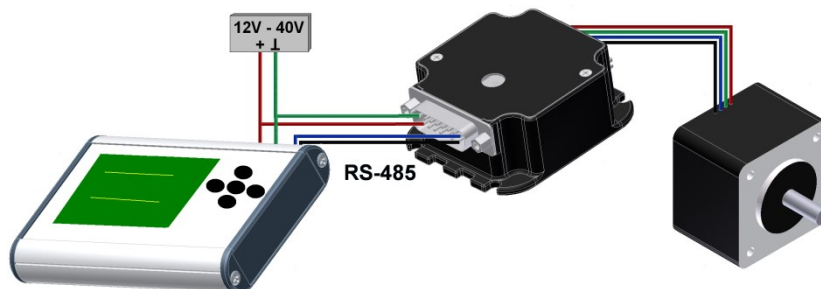
Odpověď	Hex	Překlad
/0'□	0x60	Příkaz ukončen
/0@□	0x40	Správný příkaz
/0C□	0x43	Příkazové hodnoty mimo rozsah
/0b□	0x62	Špatný příkaz
/0O□	0x4F	Přetečení

Tato tabulka však neplatí pro dotazy „QUERY COMMANDS“. Tam je odpověď trochu odlišná. Popis odpovědi na dotazový příkaz je vcelku obsáhlý a zabral by zde zbytečně moc prostoru. Na stránkách výrobce je tato problematika velmi dobře popsána.

## 4. Realizace

### 4.1. Rozbor zadání

Výsledkem této práce by mělo být zařízení, které nahradí PC nebo notebook při zadávání příkazů pro kontrolér. Příkazy se skládají z ASCII znaků a komunikace s kontrolérem probíhá po lince RS485. Důvodem náhrady PC nebo notebooku je jejich velikost a neforemnost. Samotný kontrolér může být v praxi umístěn buď přímo na motoru, nebo v jeho blízkosti a tím pádem nemusí být dobře přístupný. Pro závěrečné ladění programu mohou být rozměry zařízení překážkou.



Obr. 35 Předběžný návrh nového zařízení

Základní požadavky:

1. Malé, kompaktní rozměry
2. Zařízení musí umět komunikovat po RS485
3. Zařízení musí být schopné vytvořit a odeslat krátký řetězec složený z ASCII znaků

Další vlastnosti navrhovaného zařízení je možné odvodit z údajů ovládaných zařízení. Například komunikační rychlost, rozsah napájecího napětí, atd. Inspirací pro toto zařízení by měl být i ovládací program, kterým se nastavuje a ovládá kontrolér. Důležité je také snadná obsluha. Sestavení ASCII řetězce a jeho následné odeslání by se co do jednoduchosti mělo přiblížit zadávání z PC. Dále se musí zjistit, v jakém rozsahu ASCII tabulky se bude pracovat.

4. Komunikační rychlost 9600 nebo volitelná
5. Napájecí napětí nejlépe v rozsahu 12V až 40V stejnosměrných
6. Snadná obsluha

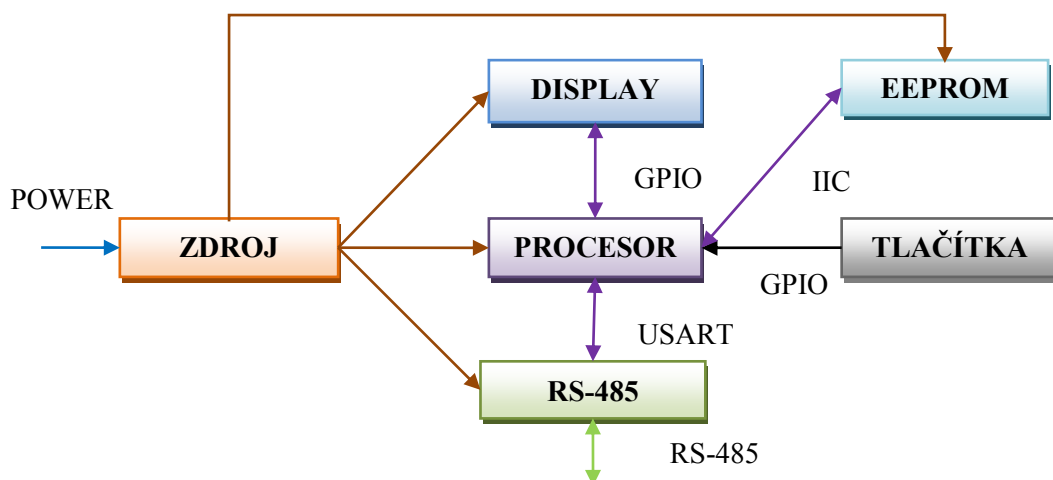
K realizaci bude nutné použít nějaký jednočipový procesor. Samotný výběr provedeme později. Dále se musí vyřešit komunikační část, zvážit použití nějaké externí paměti typu EEPROM pro ukládání nastavení a podobně, vybrat displej a také způsob jakým bude celé zařízení ovládáno. To je počet tlačítek a jejich rozmístění.

Mimo jiné by se měla řešit i dostupnost součástek, jejich cena a možnost opakované výroby. I když je jasné, že se nebude jednat o žádnou sériovou výrobu. Další výhodou je možnost úpravy programu. A to nejlépe tak, že se krabička nebude rozebírat.

7. Možnost snadné pozdější editace programu

Rozborem jsme získali sedm základních bodů, kterých se budeme nadále držet během realizace navrhovaného zařízení. Z těchto bodů je možné vycházet při výběru jednotlivých součástek, které budou v zařízení použity. Je například jasné, že procesor by měl mít asynchronní komunikaci, že zdroj by měl vydržet 40V a tak dále.

## 4.2. Volba procesoru



Obr. 36 Předběžné blokové schéma zobrazující základní části navrhovaného zařízení

V současné době se nabízí široká škála jednočipových procesorů, které jsou si na první pohled podobné, přesto se však liší. Klíčem k výběru by měla být kromě hardwarových požadavků také dostupnost procesoru, přijatelná cena, ale i vhodné vývojové prostředí. Volba padla na procesor firmy Atmel AT32UC3B1256.

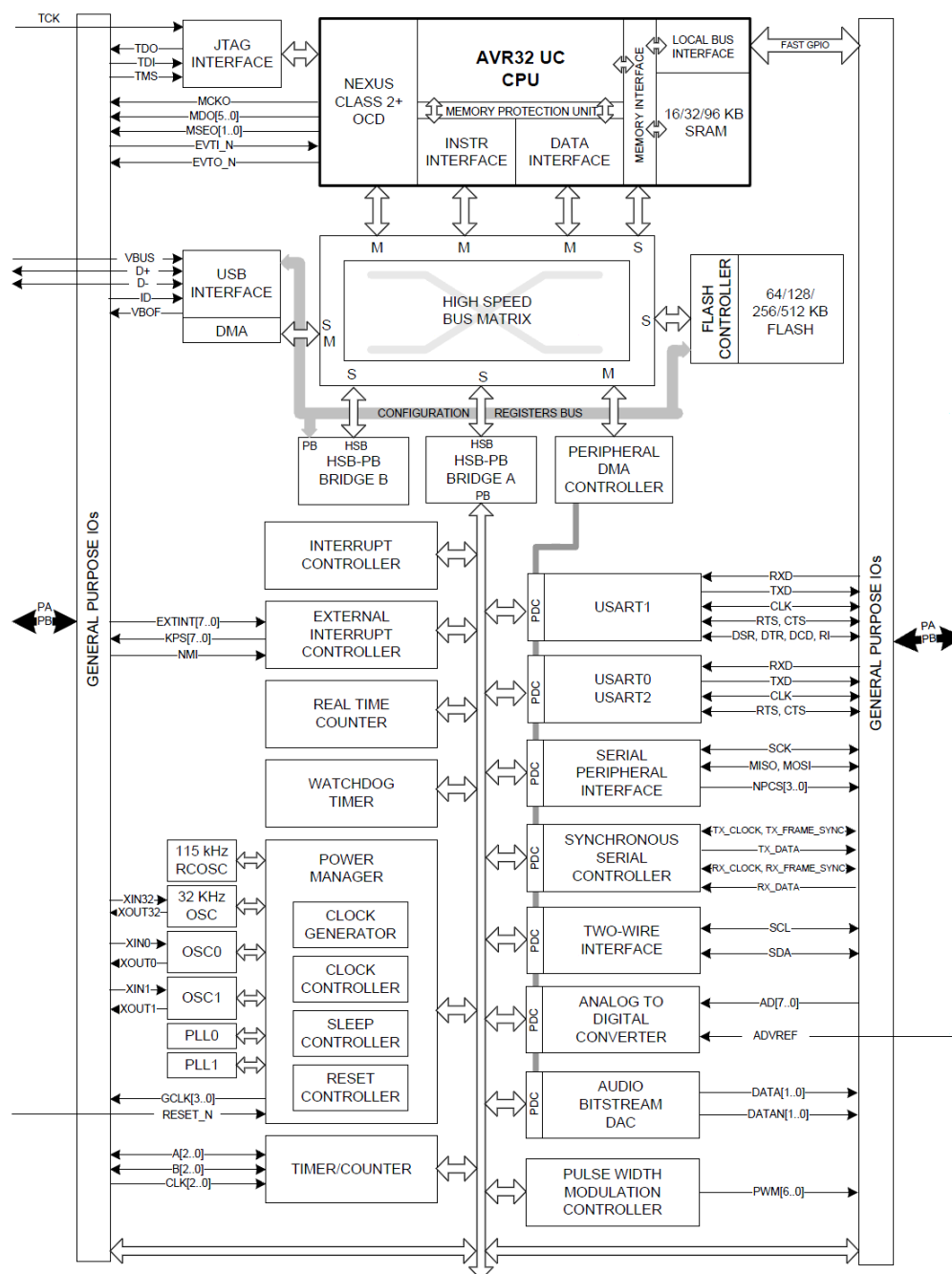
Proč zrovna tento typ? Atmel má zdarma vývojové prostředí bez omezení velikosti programu. Vývojové prostředí je trochu problém u procesorů firmy Freescale. Jejichž zdarma nabízené vývojové prostředí bývá omezené velikostí programu. Také sehnatelnou procesorů Freescale v malých sériích je špatná. Jinak jsou tyto procesory velmi dobré a kvalitní.

To, že byl vybrán procesor od firmy Atmel ovlivnily i předchozí zkušenosti s tímto typem. Po hardwarové stránce je tento čip dostačující s velkou rezervou.

*Některé parametry vybraného procesoru:*

Typ	AT32UC3B1256
Šířka slova	32 bites
Napájecí napětí	3,0V - 3,6V
Paměť Flash	256 kBytes
SRAM	32 kBytes
Počet pinů	48
Maximální počet I/O	28
Maximální frekvence (krystal)	16MHz
Maximální frekvence (PLL)	60MHz
Rozhraní	I2C, USART, SPI, USB(device)
ADC	6 x 10bitů rozlišení
Timers	10
Debug	JTAG

Nevýhodou u tohoto procesoru je absence vnitřní EEPROM a druhým mínusem je malý počet vstupů a výstupů. Přesně tento typ procesoru se však vyrábí i v 64 pinovém pouzdře. Kompatibilní typy jsou i ve velkých 100 a 144 pinových pouzdrech, takže je možné improvizovat. Jako paměť EEPROM se bude muset přidat samostatný čip připojený sériovou komunikací I2C.

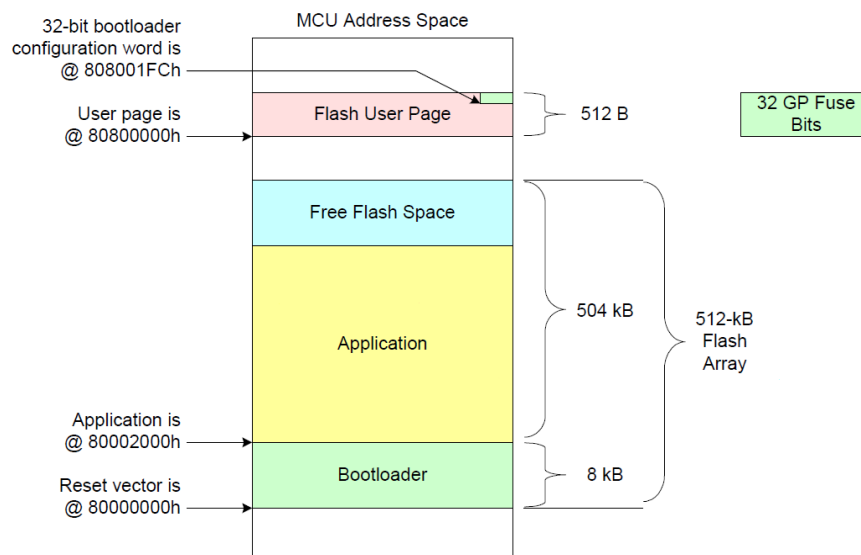


Obr. 37 Struktura procesoru AT32UC3B1256 [18]

Výhodou tohoto procesoru je možnost programování pomocí USB linky. Na stránkách výrobce je zdarma ke stažení program FLIP, který pomocí příkazového souboru s příponou „bat“ dokáže poměrně rychle naprogramovat daný procesor. S instalací tohoto programu se čas od času mohou vyskytnout menší problémy. Avšak po prostudování návodu se tyto problémy dají odstranit.

Z předchozích řádků vyplývá, že procesor obsahuje již z výroby bootloader. To je dnes vcelku běžná a užitečná věc. Musí se však dávat pozor na to, aby se tento bootloader nesmazal. Pak by se tato část programu musela do procesoru znova nahrát přes JTAG rozhraní. To však na desce není vyvedeno, protože se nevešlo.

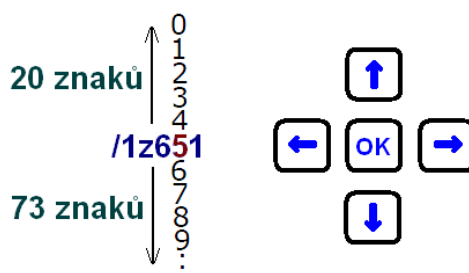
Ke smazání bootloaderu by mohlo dojít tak, že bude špatně nastavený překladač ve vývojovém prostředí. Při překladu programu z jazyka C, ve kterém bude program napsán, se musí pohlídat výsledné obsazení paměti. Bootloader je umístěn v prvních osmi kB paměti programu. Nezabírá však celou tuto velikost. Na následujícím obrázku je popis vnitřní paměti procesoru.



Obr. 38 Struktura paměti a umístění bootloADERu v procesoru AT32UC3B1256 [18]

### 4.3. Volba displeje

Asi nejdůležitější součástí navrhovaného zařízení je displej. Proto byla jeho výběru věnována značná pozornost. Displej určuje nejen vzhled celého zařízení, ale jeho vlastnosti mohou také do značné míry usnadnit ovládání zařízení. Výběr byl limitován cenou displeje, dále pak jednoduchostí obsluhy a velikostí. Postupně bylo upuštěno od znakového displeje, později od monochromatického grafického. Ten by sice vyhovoval, ale zadávání znaků pomocí křížového ovladače se hned při prvních testech jevílo jako dosti zdoluhavé. Na následujícím obrázku je příklad zadávání textového řetězce pomocí křížového ovladače. Každé jednotlivé písmenko je nutné vybrat z celého ASCII seznamu. Pokud se podíváme na znak „1“ a „z“, je mezi nimi přibližně 72 znaků. Takové zadávání může být obtížné. Řešením je použít buď větší klávesnici, nebo dotykový displej.



Obr. 39 Příklad zadávání ASCII řetězce pomocí křížového ovladače

Barevné grafické displeje s dotykovou vrstvou (tzv. touch screen) se zdály drahé a složité na ovládání. Bylo by nutné použít rychlý jednočipový procesor. Nejlépe signálový. To by se asi prodražilo, neboť je nutné započítat i vývojové prostředí.

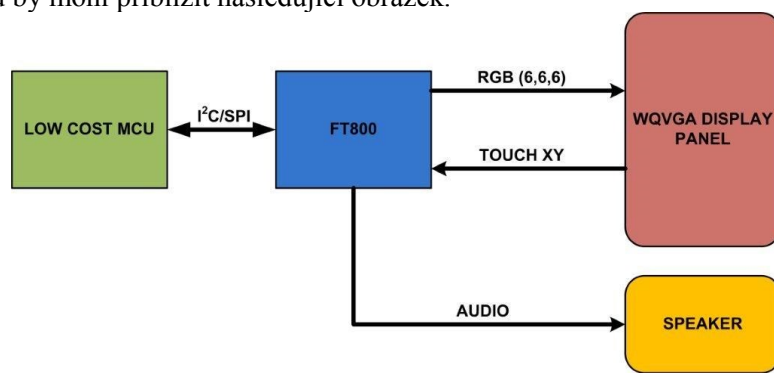
Je nutné přiznat i jisté obavy z nejistého výsledku, neboť použít neznámý procesor, neznámé prostředí a neznámý displej je velmi náročné na čas a celá aplikace by se mohla zvrtnout do sféry vzdušných zámků a nedokončených snů. Proto byl zvolen kompromis a mezi laciný procesor a displej byl vložen ovladač pro displej. Díky tomu je možné použít běžný jednočipový procesor s barevným dotykovým displejem. Zároveň se splní původní záměr naučit se něco nového a zvýší se tím pravděpodobnost, že se dá tato aplikace zvládnout.



Různí výrobci integrovaných obvodů začali nabízet čipy, které se zařadí mezi procesor a displej a působí jako ovladače pro grafické TFT displeje. Výrobci těchto integrovaných obvodů mají na svých stránkách bohatou softwarovou podporu. Tyto programy a rutiny jsou sice napsány v jazyku C nebo C++, ale většinou je nelze přímo zkopírovat do naší aplikace. Musí se trochu upravit, zjednodušit, nebo přepsat. I tak je to velmi dobrá pomoc hlavně při překonávání prvních krůčků, kdy zdánlivě nic nefunguje a výsledek je značně nejistý.

Po použití tohoto čipu zvládne i podprůměrný programátor a průměrný jednočipový procesor ovládat grafický dotykový displej. Přitom nemusí jet takzvaně na doraz a nemusí se počítat každý cyklus tak, aby displej neblíkal a zároveň dostatečně rychle reagoval.

Jak již bylo zmíněno, je více výrobců, kteří nabízejí podobný typ integrovaného obvodu. Tento takzvaný „multimediální ovladač“ byl vyvinut pro mobilní telefony nebo herní zařízení. Horší je to s jejich dostupností. I tak jsou v této době na našem trhu dostupné minimálně dva typy integrovaných obvodů, které je možné použít. Ovladače jsou nabízeny jednak v podobě integrovaných obvodů, ale i ve formě vývojových kitů, které jsou vcelku snadno použitelné. Volba padla na integrovaný obvod FT800, jehož výrobcem je firma Future Technology Devices International Limited. Funkci tohoto integrovaného obvodu by mohl přiblížit následující obrázek.



Obr. 40 Základní blokové schéma integrovaného obvodu FT800 [21]

Jak je z obrázku patrné, celý displej včetně dotykového panelu a zvukového zařízení je možné ovládat pomocí komunikace I²C, nebo sériové komunikace standardu SPI.

První typ zmíněné sériové komunikace je použit v našem zařízení pro obsluhu paměti EEPROM. Bylo by sice možné připojit displej do série, ale SPI se zdá být pro obsluhu displeje rychlejší a jednodušší. Hardwarový ovladač pro SPI je v procesoru implementován, takže se jeho využití přímo nabízí.

Pro první ladění byla zakoupena přímo vývojová deska od firmy FTDI VM800C35A-D. S vývojem programu se mohlo začít dříve, než byla hotová finální deska. To ušetřilo hodně času. Celý kit je jen o něco málo dražší než samotný displej. Displej se dá z vývojového kitu snadno odpojit a použít ve finální aplikaci.



Obr. 41 Vývojová deska VM800C35A-D [21]

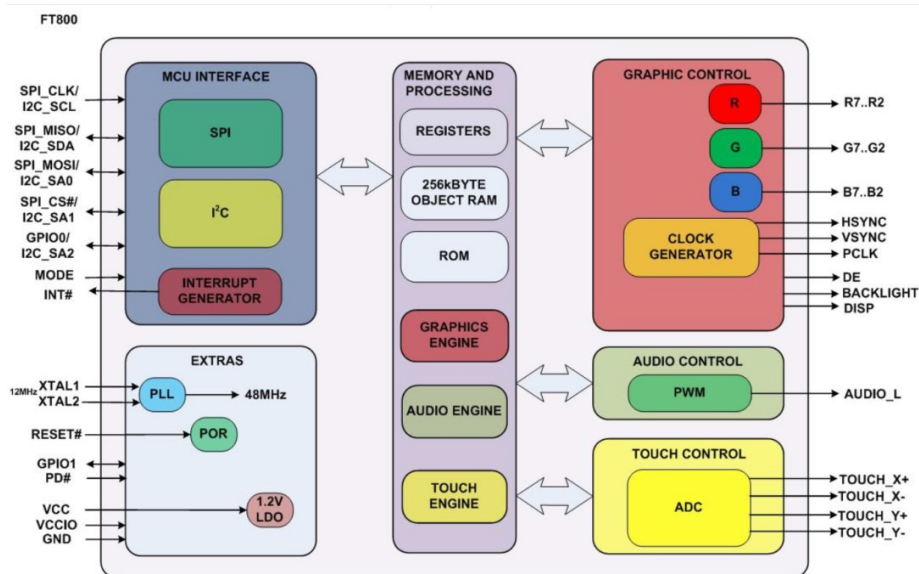
## 4.4. Popis obvodu FT800

Základní parametry obvodu:

- Pracuje s QVGA a WQVGA TFT displeji s paralelním RGB interfejsem
- Výstup pro barvy 3 x 6 bitů
- 256k interní paměti pro objekty
- Uloží až 2000 objektů
- Vestavěný kontrolér podporuje čtyřvodičové rezistivní dotykové panely
- PWM výstup pro řízení jasu
- Anti-aliasing vylepšující zobrazení
- Nízká spotřeba - 35mA v aktivním módu a 25μA v uspaném stavu
- Widgety umožňují jednoduchou tvorbu komplexních objektů (např. analogové hodiny)
- Integrovaný audio výstup
- SPI nebo I<sup>2</sup>C interfejs k připojení k mikrokontroléru
- 48LD QFN pouzdro [21]

Již na první pohled je patrné, že tento obvod velmi usnadní celou aplikaci. Ovládání TFT barevného displeje s paralelním rozhraním klade na zařízení značné nároky na rychlost, ale i na paměť. To do značné míry omezuje použití běžných a laciných jednočipových procesorů. Integrovaný obvod FT800 dokáže tyto nároky převzít na sebe, proto tyto procesory takřka jic vrací do hry.

Pro rychlou obsluhu TFT displeje nutné při překreslení obrázků přenést značné množství dat do displeje během velmi krátké doby. Taktovací frekvence je nad 25MHz a jeden obrázek o rozlišení 360x240 bodů při šesti bitech na každou barvu má velikost kolem 195kB.



Obr. 42 Blokové schéma integrovaného obvodu FT800

Obvod FT800 má vnitřní paměť do které se obrázek poskládá a potom se celý vyšle na displej. I tak by bylo docela náročné připravit obrázek pomocí sériové komunikace SPI. Proto je pro uživatele připraveno několik různých pomůcek. Vcelku jednoduše je možné vytvořit čáry, obdélníky nebo kruhy. V pevné paměti integrovaného obvodu je připraveno několik sad znaků různých velikostí a různého vzhledu. Díky tomu je možné poměrně rychle připravit novou stránku displeje.

Aby byl popis přesnější, je nutné uvést, že integrovaný obvod FT800 se skládá z několika částí. Celý obvod je taktován vnitřní časovou základnou, která má vlastní oscilátor řízený krystalem. Rychlost krystalu je 12MHz. Obvod však obsahuje PLL smyčku, kterou je možné během inicializace zapnout. PLL smyčka se dnes používá v mnoha aplikacích. Důvodem jsou vyšší rychlosti jednočipových procesorů, ale i ostatních zařízení. Krystaly kmitají na základních frekvencích

takzvanými střížnými kmity pouze v oblasti do cca 20Hz. Uvádí se, že je možné vyrobit základní kmitočty do 40MHz, ale náročnost výroby těchto součástek zvyšuje jejich cenu. Běžné krystaly kmitající v oblasti nad cca 16 MHz využívají třetí nebo pátou harmonickou. Tady je však těžké zaručit stabilní funkci oscilátoru (a hlavně rozběh) při dodržení nízké spotřeby. PLL smyčka se tak jeví jako velmi dobré a přesné řešení pro zvýšení taktovací frekvence.

Ale vraťme se k popisovanému obvodu. Zapnutím PLL smyčky po inicializaci je možné přejít na vnitřní kmitočet 48MHz. Tím se zvýší nejen rychlost vykonávání jednotlivých příkazů, ale může se též zvýšit rychlost komunikace. U sériové SPI komunikace se je pak hodinový kmitočet možné zvýšit až na 30MHz a u I<sup>2</sup>C komunikace je možné zvýšit kmitočet na 3,4MHz.

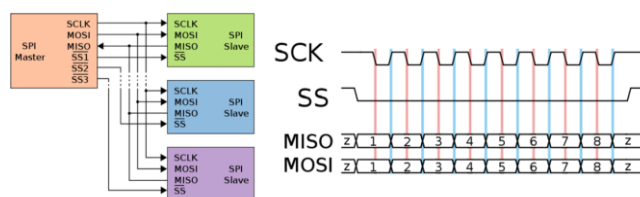
Další důležitou částí obvodu FT800 je komunikační rozhraní. Zde jsou nabídnuty dva nejpoužívanější standardy sériové komunikace.

Prvním standardem je typ je I<sup>2</sup>C. Tato komunikační sběrnice bude podrobněji popsána u sériové paměti EEPROM.

Druhým standardem je komunikace SPI. Název tvoří počáteční písmena anglických slov Serial Peripheral Interface. Jedná se o třívodičovou obousměrnou datovou komunikaci, která je tvořena hodinovým signálem, a dvěma datovými signály MOSI (Master Output Slave Input) a MISO (Master Input Slave Output). Dále se používají samostatné vodiče se signály CS (resp. SS). Tyto signály jsou aktivní v nule a aktivují jednotlivá zařízení.

Na lince je připojen vždy jen jeden master, ale může být připojeno několik zařízení jako slave. Slave zařízení, se kterým master komunikuje je potom připojeno signálem CS v logické nule. Ostatní zařízení mají tento signál v logické jedničce. Data na lince se posílají zároveň oběma směry. Hodiny jsou společné. Platnost jednotlivých bitů je dána náběžnou nebo sestupnou hranou.

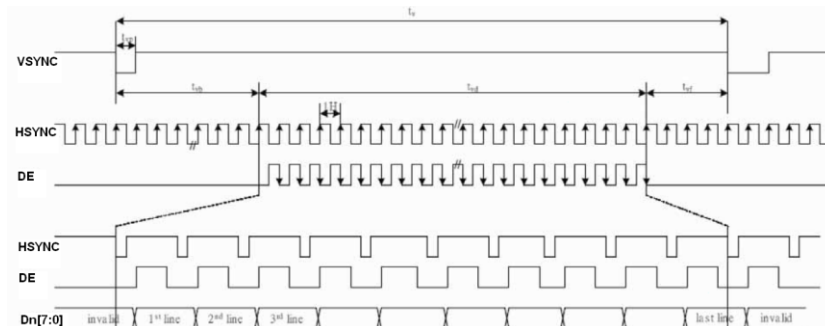
Komunikace SPI se používá v mnoha aplikacích. Například pro paměťové karty typu SD nebo MMC, nebo pro komunikace s AD, DA převodníky atd.



Obr. 43 SPI komunikace [30]

Dalším blokem ve struktuře integrovaného obvodu FT800 je blok grafický. Ten má za úkol vyrábět signály pro TFT displej. Parametry těchto signálů jsou nastavitelné a mění se pro jednotlivé typy displejů podle jejich rozlišení.

Jen pro představu, je zde možné uvést, že displej potřebuje hodinový signál, dále signály horizontální a vertikální synchronizace a v neposlední řadě data, která určují barvy jednotlivých pixelů. Každý bod má tři barvy po 6 bitech. V následujícím obrázku je znázorněno časování jednotlivých signálů. Aby byla představa o rychlosti, mělo by se uvést, že clock se pohybuje kolem 25MHz.



Obr. 44 Signály generované pro displej [22]

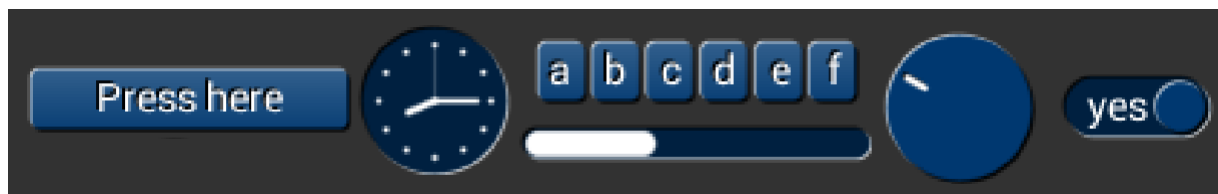
Struktura čipu FT800 obsahuje dále audio-obvod, který umožňuje generovat různé zvuky. Obsluha je velmi snadná. Nabídka jednotlivých zvuků je pestrá. Obsahuje možnost volby různých pípnutí a sirén, ale taky kompletní stupnici. Navíc se může měnit tón podle hudebního nástroje. Integrovaný obvod nabízí i možnost přehrání audio-souboru.

Ovládání zvukového obvodu je snadné. Jednoduše je možné ovládat tóny i hlasitost. Výstup si proto zaslouží, aby byl osazen zesilovačem a miniaturním reproduktorem.

Dotykový panel odporového typu je obsluhován pomocí dvou desetibitových převodníků. Díky vnitřním algoritmům není nutné se o tyto převody starat a je možné snadno přečíst polohu dotyku. Integrovaný obvod nabízí i různé varianty pro čtení této polohy a také možnost nastavení výstupního pinu, který vyvolá přerušení u nadřazeného procesoru.

V nabídce je vypnutí obsluhy dotykového panelu, jednorázové přečtení dotykového panelu, kontinuální čtení synchronizované s obrazovou frekvencí obrazovky (60Hz) nebo kontinuální provoz, kdy je frekvence snímání přibližně 1000 převodů za sekundu. Odezva na dotyk je tím pádem dostatečně rychlá. V integrovaném obvodu je navíc uložena rutina pro kalibraci dotykového panelu, kterou je možné vyvolat.

Integrovaný obvod obsahuje i jakýsi grafický koprocessor, který umožňuje pomocí několika málo příkazů vyvolat obsáhlejší implementované grafické rutiny. Takto je možné vyvolat například takzvané widgety pro zobrazení analogových hodin, trojrozměrná tlačítka, přepínače, klávesnici a mnoho dalších užitečných věcí.



Obr. 45 Příklady některých „widgetů“, které nabízí FT800 [21]

Paměť integrovaného obvodu FT800 je rozdělena na několik částí. První velkou část zabírá přímo grafická RAM. Dále následují identifikační údaje. Za nimi jsou fonty. Jednak jsou zde pevné fonty o několika velikostech a pak je zde RAM, kde je možné uložení vlastních fontů.

Protože znaky některých fontů nejsou stejně dlouhé, je v následující části uložena informace o tom kolik obrazových bodů který znak zabírá.

Prostor označený jako RAM\_DL slouží k vytvoření vlastního displeje podle základních příkazů. Tato část paměti se plní jako FIFO a na stránkách výrobce je zobrazena jako kruh. Při zápisu nového příkazu se musí program vždy přesvědčit, zda je již paměť uvolněná. Při větších aplikacích se může lehce přepsat část, kterou obvod FT800 ještě nezpracoval.

Tab. 10 Rozdělení adresového prostoru integrovaného obvodu FT800

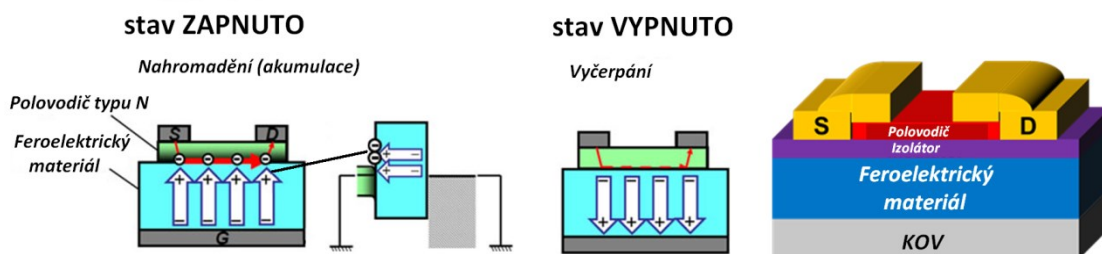
Počáteční adresa	Konečná adresa	Velikost	Název	Popis
0x000000	0x03FFFF	256 kB	RAM_G	Hlavní grafická RAM
0x0C0000	0x0C0003	4 B	ROM_CHIPID	Identifikační údaje
0x0BB23C	0x0FFFFB	275 kB	ROM_FONT	Tabulky fontů a bitmapy
0x0FFFFC	0x0FFFFF	4 B	ROM_FONT_ADDR	Adresy a délky znaků fontů
0x100000	0x101FFF	8 kB	RAM_DL	Display List RAM
0x102000	0x1023FF	1 kB	RAM_PAL	Palette RAM
0x102400	0x10257F	380 B	REG_*	Registry
0x108000	0x108FFF	4 kB	RAM_CMD	Vyrovňovací buffer pro grafický koprocessor

Sice to tak na první pohled nevypadá, ale tento popis obvodu FT800 je dosti stručný a zmiňuje jen povrchně funkce daného čipu. Podrobný popis poskytuje výrobce na svých stránkách. Ovládání displeje a tvorba jednotlivých obrazů bude dále popsána v samostatné kapitole.

#### 4.5. Paměť EEPROM (FRAM)

Je sice možné využít k ukládání různých dat programovou paměť FLASH procesoru, ale osobně se mi tato možnost nelíbí z důvodu nebezpečí poškození programu. Také mazání této paměti po blocích není zrovna ideální. Proto se v zařízení použije externí sériová paměť EEPROM s komunikací I<sup>2</sup>C (Inter-Integrated Circuit). Vybrán byl typ 24LC256, který by měl svojí velikostí paměťového prostoru vyhovovat budoucím požadavkům. Při pozdějším ladění byl však tento typ vyměněný za jiný. Kapacita byla sice dostatečná, až moc velká, ale počet zápisů se mi zdál malý. U paměti EEPROM je počet zápisů omezený. Vybraný typ zvládne asi jeden milion zápisů.

Původní záměr byl takový, že by se paměť EEPROM využila pouze pro uchování nastavovacích parametrů. K tomu by deklarovaný počet zápisů bohatě stačil. Při ladění mě však napadlo, že by se mohlo ukládat několik posledních příkazů. Sice se nepředpokládá, že by bylo využití zařízení tak vysoké, ale pro jistotu jsem změnil typ paměti z EEPROM na FRAM. Cenově to nebyl výrazný rozdíl, protože se značně snížila kapacita paměťového prostoru paměti. Počet zápisů se zvýšil minimálně tisíckrát, což mnohonásobně přesáhne životnost zařízení. Pro dané účely 2kB paměti budou stačit.



Obr. 46 Základní princip paměti FRAM [23]

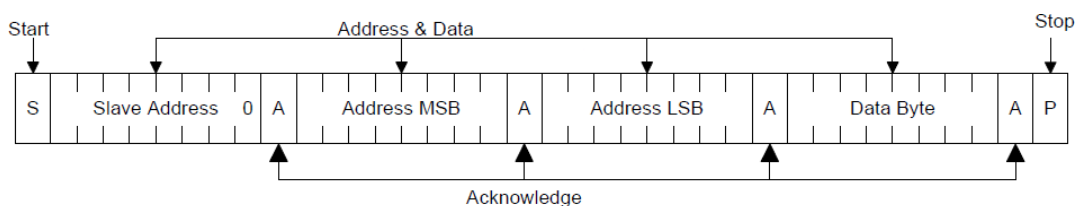
Krátce k principu paměti FRAM. Buňky paměti FRAM využívají zbytkové polarizace feroelektrického materiálu. Princip fungování je založen na vystavení elektrického krystalu elektrickému poli. Díky tomu se centrální atom pohne ve směru elektrického pole a zapříčiní proudový náraz. Ten je zaznamenán vnitřními obvody a paměť je následně zapnuta. Díky tomu, že centrální atom v této pozici zůstane i po odstranění pole, je paměťový stav uchován. [24]

*Porovnání vlastností vybraných typů paměti:*

Typ	24LC256	FM24CL16
Napájecí napětí	2,5V až 5,5V	2,5V až 5,5V
Maximální rychlost signálu „clock“	400kHz	1 MHz
Velikost paměťového prostoru	256 kBite $\approx$ 32 kByte	2 kByte
Max. doba zápisu	5ms	NoDelay™ Writes
Počet vymazání/zápis	1 000 000	unlimited
Odběr (aktive SCL=400kHz)	400uA (3mA write)	150uA

Sériové paměti s I<sup>2</sup>C komunikačním protokolem se mohou lišit způsobem adresování uchovaných dat. Pro menší paměti se používá pouze jeden byte adresy, pro větší se používají buď dva byte za sebou, nebo se někdy využívají i nižší bity z „chip select bits“. Při ladění aplikace se stává, že po výměně paměti za typ s větší velikostí paměťového prostoru, přestane program fungovat. Nebo data přečtená z paměti nesouhlasí s daty uloženými. Příčinou často bývá právě rozdílné adresování paměti.

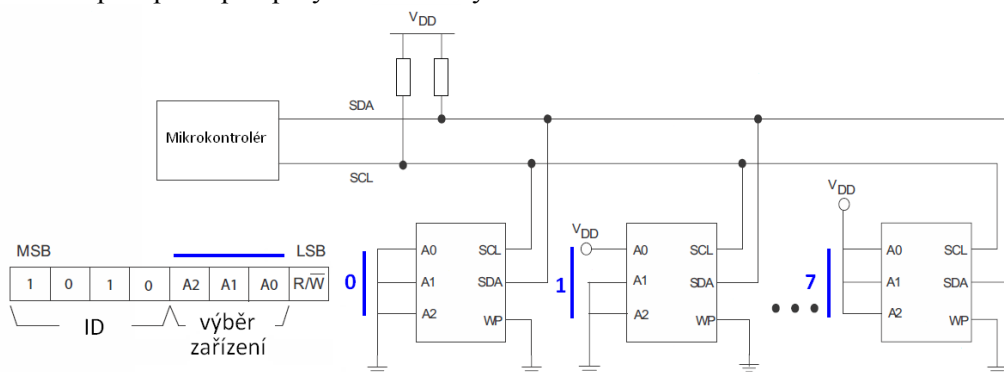




Obr. 47 Adresování použité paměti [25]

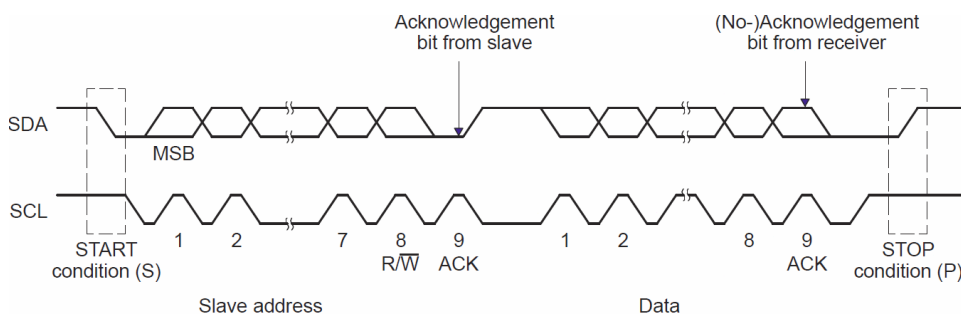
Komunikační protokol I<sup>2</sup>C byl vyvinut v osmdesátých letech minulého století firmou Philips pro obousměrnou komunikaci mezi jednotlivými integrovanými obvody pomocí dvou drátů. V takovéto síti je jeden master a několik zařízení slave. Na jedné lince je pak maximálně 128 různých propojených zařízení.

Jeden vodič je ovládaný pouze mastrem a slouží jako hodiny. Druhý vodič je datový a je takzvaně tahán přes pull up odpory k zemi různými zařízeními.



Obr. 48 IIC komunikace s použitím HW adresy [24]

Přenos dat začíná a končí specifickou posloupností přepnutí datového a hodinového signálu. Tato posloupnost se nesmí vyskytnout nikde jinde v přenosu dat. Tímto je stanoven startbit a stopbit. Další zvláštností jsou takzvané ACK bity, kterým zařízení nebo master potvrzují přijatá data.



Obr. 49 IIC komunikace startbit a stopbit [25]

Asi je zbytečné kopírovat sem sekvenci zápisu a čtení. Je to v každém popisu paměti. Zatímco pro zápis se posílá „control byte“ pouze jednou s nulovým bitem R/W (za ním následuje adresa a pak data), pro čtení z vybrané adresy se musí poslat „control byte“ dvakrát. Jednou s nulovým R/W pro zápis adresy a za touto adresou se posílá „control byte“ podruhé s R/W v jedničce. Poté už se jen čtou následující data.

Dnes se však již tato komunikace ovládá přes hardwarové rozhraní implementované v procesoru, a tak se tyto podrobnosti moc neřeší. Bývaly však doby, kdy při ladění software dokázala tato komunikace docela dost potrápit.

#### 4.6. Zdrojová část

Na první pohled vypadá zdroj jako jednoduchá záležitost, ale opak je pravdou. Navrhované zařízení bude pravděpodobně napájeno společně s kontrolérem, ke kterému bude připojeno. Nebylo by praktické připojovat zvláštní zdroj.

Aby zařízení fungovalo v celém rozsahu napájecího napětí, které zvládne kontrolér, je nutné dobře spočítat ztráty. Největší problém bude pravděpodobně podsvětlení displeje, které má ze všech obvodů nejvyšší odběr. Rozložení napětí na desce naznačuje následující tabulka:

Tab. 11 Tabulka napěťových úrovní na navrhované desce

Popis napětí	Hodnota [V]
Vstupní napětí maximální	40
Napětí pro obvody RS485	5
Napětí na procesoru, paměti a obvodu FT800	3,3
Podsvětlení displeje	cca 5 - 10

Pomocí dokumentace od výrobců součástek je možné přibližně určit maximální odběr celého zařízení. Jednotlivé proudy jsou zaznamenány v Tab. 12.

Tab. 12 Tabulka přibližného odběru součástek

Součástka	Odběr [mA]
24LC256	0,4
AT32UC3B1256	24 (aktive 60MHz)
75176	Příjem 8 / Vysílání 60
FT800	20
Displej	10
Podsvětlení	max. 20
Zvukový obvod	0,1 / v činnosti cca 80
<b>Celkem</b>	85,5 (bez zvuku a vysílání)

Zvuk se aktivuje na cca 0,6s při stisknutí tlačítka. Vysílání je aktivní na 5 – 250ms. Podsvětlení je možné napájet zvlášť z proudového zdroje. Tím se sníží klidový proud na 65,5mA.

Při maximálním vstupním napětí  $U_{\max} = 40V$  a výstupním napětí  $U_{\text{out}} = 5V$  bude na stabilizátoru ztrátový výkon 2,28W. Na stabilizátoru proudového zdroje bude ztrátový výkon 0,7W. To platí pro lineární stabilizátory. Spínané zdroje na tom jsou lépe.

$$P = U \cdot I \quad (8)$$

$$P_1 = 35 \cdot 0,065 = \underline{2,28W}$$

$$P_2 = 35 \cdot 0,020 = \underline{0,7W}$$

$$R_{th(j-a)} = \frac{\vartheta_a - \vartheta_j}{P} = \frac{\Delta\vartheta}{P} \quad (9)$$

$$R_{th(j-a)} = R_{th(j-c)} + R_{th(c-s)} + R_{th(s-a)} \quad (10)$$

$$\Delta T_{(j-a)} = R_{th(j-a)} \cdot P_{tot} \quad (11)$$

Aby se zjistilo, zda se stabilizátor uchladí, musí se vypočítat teplotní odpor stabilizátoru. Výpočtem by se mělo dále zjistit, zda se čip stabilizátoru neohřeje nad kritickou mez, která je dána výrobcem. Její hodnota pro LM317 je  $T_j < 125^\circ C$ .

Pro napájení bychom mohli použít stabilizátor LM317 HVT, který má vyšší maximální vstupní napětí. Jeho parametry jsou následující:

$$U_{in\_max} = 57 \text{ V}$$

$$R_{th(j-c)} = 4 \text{ K/W}$$

$$R_{th(c-s)} = 0,2 \text{ K/W}$$

$$R_{th(c-a)} = 50 \text{ K/W bez chladiče}$$

Chladič o rozměrech 17 x 13,5 x 9 mm, který je v nabídce GM electronic pod názvem DO2A má teplotní odpor:

$$R_{th(s-a)} = 22 \text{ K/W}$$

Výpočet celkového teplotního odporu:

$$R_{th(j-a)} = R_{th(j-c)} + R_{th(c-s)} + R_{th(s-a)} = 4 + 0,2 + 22 = \underline{26,2 \text{ K/W}}$$

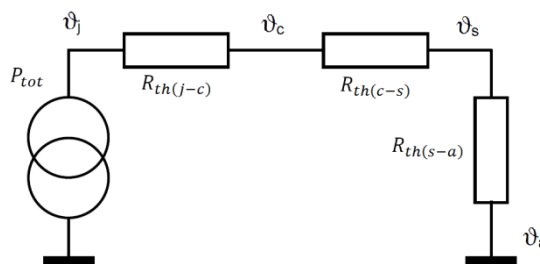
Výpočet rozdílu teplot čip – okolí:

$$\Delta T_{(j-a)} = R_{th(j-a)} \cdot P_{tot} = 26,2 \cdot 2,28 = \underline{59,74 \text{ K}}$$

Výpočet teploty na čipu při teplotě okolí  $T_a = 40^\circ\text{C}$ .

$$T_{(j)} = \Delta T_{(j-a)} + T_{(a)} = 59,74 + 40 = \underline{99,74 \text{ K}}$$

Stabilizátor s daným chladičem by měl vydržet tento ztrátový výkon. Musíme brát v úvahu, že napájení 40V se bude používat jen výjimečně. Na druhou stranu se odběr zvyšuje při vysílání, anebo při funkci zvukového obvodu.



Obr. 50 Tepelné schéma stabilizátoru

Stabilizátor proudového zdroje má přibližně třetinovou ztrátu, takže by měl vydržet danou ztrátu bez problémů. Duralová krabíčka by měla zabránit přehřátí vnitřního prostoru zařízení.

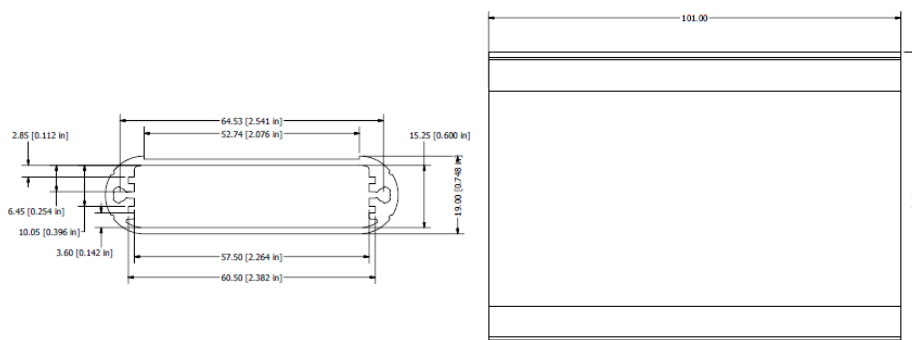
Lepších výsledků by se dosáhlo při použití spínaného zdroje. Jsou zde však obavy z rušení vnitřních obvodů displeje, a zvukového obvodu. Použitá krabíčka také do značné míry limituje velikost použitých kondenzátorů.



#### 4.7. Konstrukce přístroje

Vzhled a rozměry přístroje udává použitá krabička. Ačkoliv je na trhu hodně výrobců přístrojových krabiček, výběr není vůbec jednoduchý. Krabička by měla být pevná, odolná, malá, ale zároveň se do ní musí vejít všechny části přístroje. Z přístroje bude vyveden konektor pro připojení napájení a konektor pro komunikaci. Tyto dva konektory je možné sloučit do jednoho vícepinového. Konektor pro programování je standardní USB konektor. Rozměrově nejlépe vychází konektor USB-B mini. Je předpoklad, že program se bude dále upravovat, proto by měl být tento konektor také vyveden ven z krabičky.

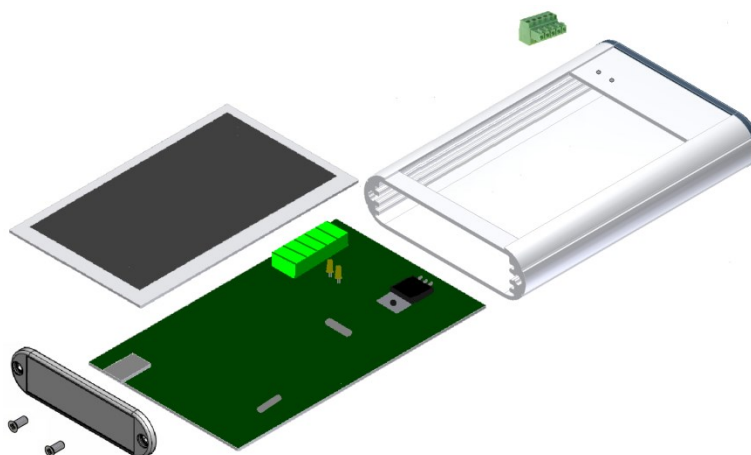
Volba padla na duralovou krabičku 1455B1002. Výrobcem je kanadská firma Hammond Manufacturing. Krabička se skládá z uzavřeného duralového profilu, který je nařezán na potřebnou délku a zakončen plastovými čely. Ta jsou z boku upevněna pomocí dvou samořezných šroubů.



Obr. 51 Rozměry použité krabičky [26]

Tato krabička má kompaktní rozměry a padne dobře do ruky. Duralový plášť je pevný a bude dobře odvádět nadbytečné teplo, které se bude tvořit uvnitř přístroje. Profil krabičky je opatřen drážkami pro uložení desky plošného spoje a displeje. Displej se do drážky bohužel nevejde, takže bude nutné vodička upravit pilníkem.

Do krabičky se také musí vypilovat obdélníkový otvor pro displej a na horní hraně otvor pro konektor. Rozmístění jednotlivých dílů nám ukazuje obrázek Obr. 52.



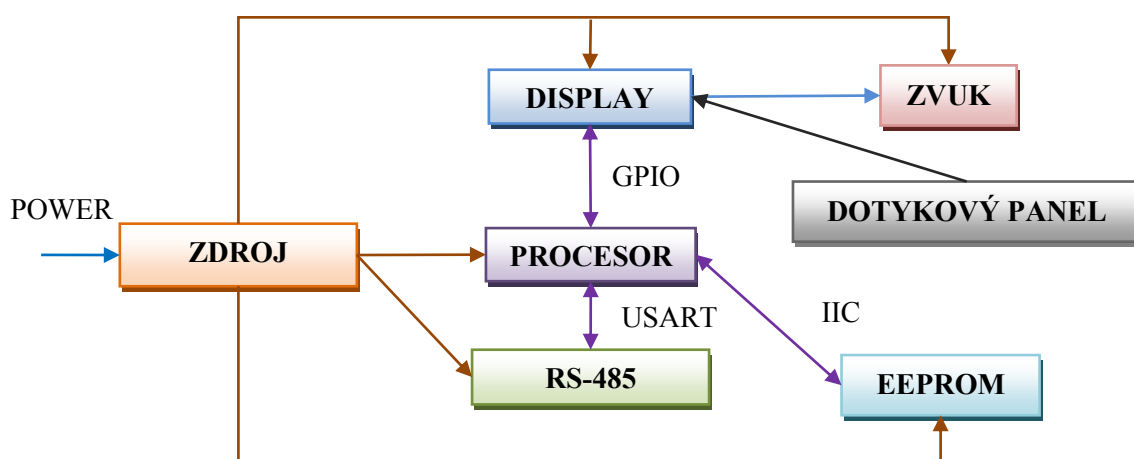
Obr. 52 Konstruktivní uspořádání výrobku

Nevýhodou krabičky je uzavřený profil, takže k plošnému spoji bude špatný přístup. Plošný spoj bude nutné zasunout zároveň s displejem tak, aby nedošlo k rozpojení propojovacího kablíku. Odhlazení obvodu musí proběhnout mimo krabičku.

## 4.8. Schéma zapojení

Pro kreslení schéma a k návrhu plošného spoje byl použit program Eagle od firmy CadSoft. Pro dvě vrstvy, malý počet součástek, velikost 80x100mm a zároveň nekomerční využití je k dispozici zdarma.

Schéma zapojení vychází z obrázku *Obr. 36*. Od tlačítek bylo díky dotykovému displeji upuštěno. Aby byl stisk panelu zřetelný, byl přidán blok se zvukem. Výsledkem je upravené blokové schéma na *Obr. 53*. Podrobné schéma je v příloze.



*Obr. 53 Upravené blokové schéma navrhovaného zařízení*

**Zdroj** byl popsán v kapitole 4.6. Schematicky byl navržen jako lineární zdroj s integrovaným obvodem LM317HVT. Je to upravená verze klasické LM317. Na vstup je možné připojit až 57V (60V max.). Výstupní napětí je nastaveno na 5V. Z tohoto zdroje se napájí přímo komunikační část s převodníkem pro RS485. Dále je z pětivoltové sítě napájen pevný stabilizátor 3,3V pro procesor, obvod FT800, a displej. Do této sítě ústí přes diodu USB konektor. Aplikace může být napájena přímo z USB konektoru. Asi nebudou funkční všechny části, ale pro programování by měl být USB zdroj dostatečný. Při přeprogramování zařízení nebude nutný externí zdroj.

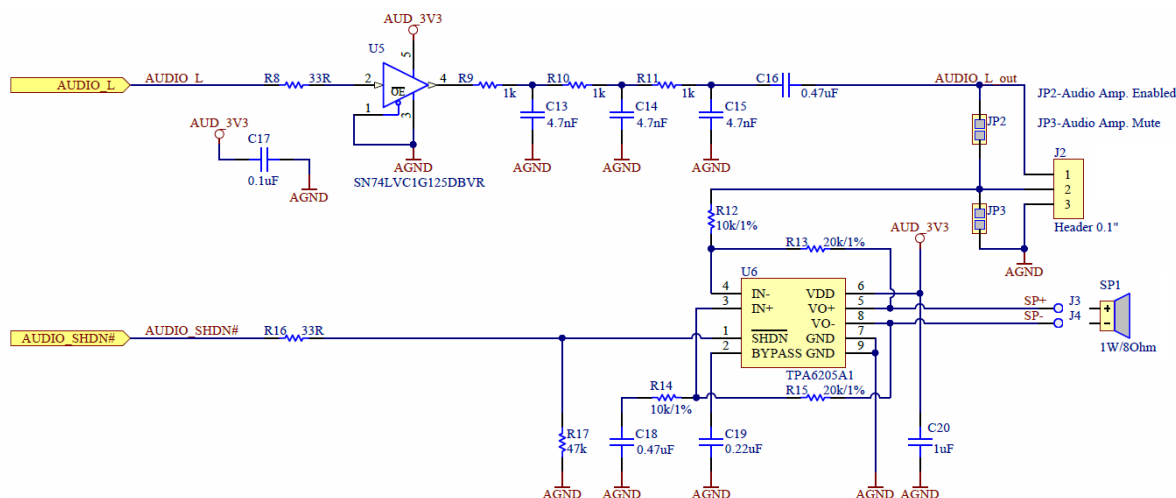
Druhý obvod LM317HVT by měl napájet podsvětlení displeje. Je zapojen jako proudový zdroj s omezením kolem 20mA. Přesná hodnota bude určena podle jasů displeje během ladění.

**Procesor** a okolí procesoru bylo navrženo podle dokumentace od výrobce. Oscilátor byl vybrán krystalem řízený o frekvenci 12MHz. Při návrhu desky se potom musí dát pozor na to, aby byl oscilátor co nejblíže procesoru a byl mimo datové vodiče, které by ho mohly rušit. Snižování spotřeby procesorů se dotklo i vnitřních zapojení oscilátorů. Oscilátory proto bývají citlivé na rozběh a rušení.

Napájecí část procesoru je filtrována kapacitami 100nF a tantalovými kondenzátory 33uF/10V. Během osazování bude zvážena změna kapacity. To podle dokumentace a polohy daného kondenzátoru. Měly by se použít i nižší kapacity například 100p nebo 4,7nF. Každá změna bude doplněna do schématu. Kondenzátory jsou rozloženy tak, aby obsáhly všechny zdroje rušení. Keramické kondenzátory jsou použity jako filtr i pro napájení jádra procesoru. Zdroj tohoto napětí je přímo v procesoru a má na výstupu napětí 1,8V.

**EEPROM** je použita v pouzdře SO8 s komunikací EEPROM. Pouzdro i zapojení je univerzální a integrovaný obvod je možné zaměnit nejen za obvod jiného výrobce, nebo za čip s jinou kapacitou paměti, ale je také možné použít úplně jiný typ paměti. Již v předchozí kapitole byla zmíněna možnost použití paměti FRAM. To, jaký čip se použije, se bude řešit až během návrhu programu.

Zvuk je v zařízení vyráběný pomocí 1,25W zesilovače a malého reproduktoru. Integrovaný obvod FT800 obsahuje ve svém zapojení i zvukový generátor. K PWM výstupu tohoto generátoru je potom přes filtr připojen již zmíněný zesilovač s reproduktorem. Schéma bylo opět převzato z dokumentace od výrobce integrovaného obvodu FT800. Pouzdro originálního integrovaného obvodu TPA6205 je velmi malé a v amatérských podmínkách těžko pájitelné. Navíc obvod nebyl ve skladech běžných dodavatelů elektronických součástek. Proto byl integrovaný obvod TPA6205 nahrazen obvodem TPA2005. Zapojení zůstalo stejné.



39

**RS485** je vytvořena z obvodu 75176. Tento obvod je v univerzálním pouzdře SMD SO8. Je možné ho nahradit jiným typem. Například MAX485, SP485 a tak dále. V nabídce je široká škála od různých výrobců. Jednotlivé typy se liší spotřebou, rychlostí, vyšší odolností proti přepětí (až 80V), nebo napájecím napětím, ale mají společné pouzdro i zapojení.

Schéma i deska jsou navrženy tak, aby bylo možné osadit tento komunikační obvod jako galvanicky oddělenou část s vlastním DC/DC konvertorem. Signál je oddělen pomocí optočlenů. Pro komunikaci jsou použity rychlé optočleny s hradlem, pro přepínání směru komunikace je osazen laciný pomalý typ s tranzistorem.

Galvanické oddělení nebude doufám potřebné, ale kdyby nastal nějaký problém, je snadnější ho osadit na předem připravenou desku než vyrábět nový plošný spoj. S největší pravděpodobností není komunikace u zařízení R356 galvanicky oddělena od zdroje, ani oddělení nebude vyžadovat.

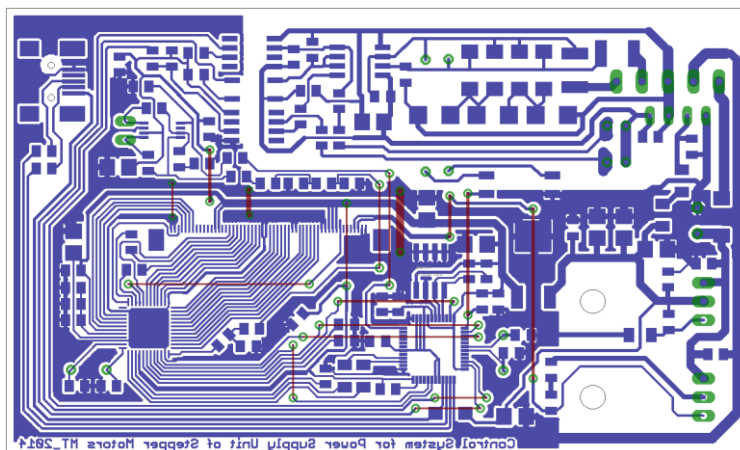
Mezi integrovaným obvodem 75176 a vstupně výstupním konektorem je vložen ochranný obvod, který by měl zamezit zničení převodníku po chybném připojení napájecího napětí na linku RS485.

Napájecí napětí pro obvod RS485 je 5V. Zbytek desky je napájen z napětí 3,3V. To působí problém při připojení převodníku na procesor. Piny procesoru, které byly vybrány pro komunikaci zvládnou 5V logiku (5V input tolerant). Připojení procesoru je potom přes oddělovací odpor 1kΩ. Za tímto odporem je potom odpor 22kΩ, který „tahá“ pin převodníku k napětí +5V. Při Logické nule na procesoru bude na pinu převodníku napětí cca 0,22V což je dostatečně nízko pod rozhodovací úroveň.

#### 4.9. Návrh plošného spoje

Jak již bylo zmíněno, schéma i plošný spoj byl navržen v programu Eagle. Spojе byly zapojeny ručně bez použití autorouteru.

Deska byla od prvopočátku koncipována jako jednostranný spoj. Po několika pokusech o vyleptání v amatérských podmínkách byla deska změněna na oboustrannou a zaslána do výroby.



Obr. 56 Původní návrh jednostranné desky s propojí (transparentní pohled ze strany TOP)

Návrh desky byl přizpůsoben vybrané krabičce a použitému displeji. To byly základní omezující pravidla, podle kterých se celý návrh odvíjel. Bylo nutné hlídat výšku součástek a jejich rozmístění.

Konstrukční třída přesnosti 6 byla dána procesorem AT32UC3B1256 (pouzdro TQFP48) s roztečí 0,5mm. Mezera mezi pájecími ploškami vyšla na 0,2mm. Rozteč 0,5mm má i konektor 54 pinů pro displej. Stejnou rozteč má také integrovaný obvod FT800 (pouzdro VQFN48). Tady byly obavy ze správného zapájení obvodu. Prodávané pouzdro nemá SMD vývody. Navíc je kontakt GND vyveden do plošky pod integrovaným obvodem.

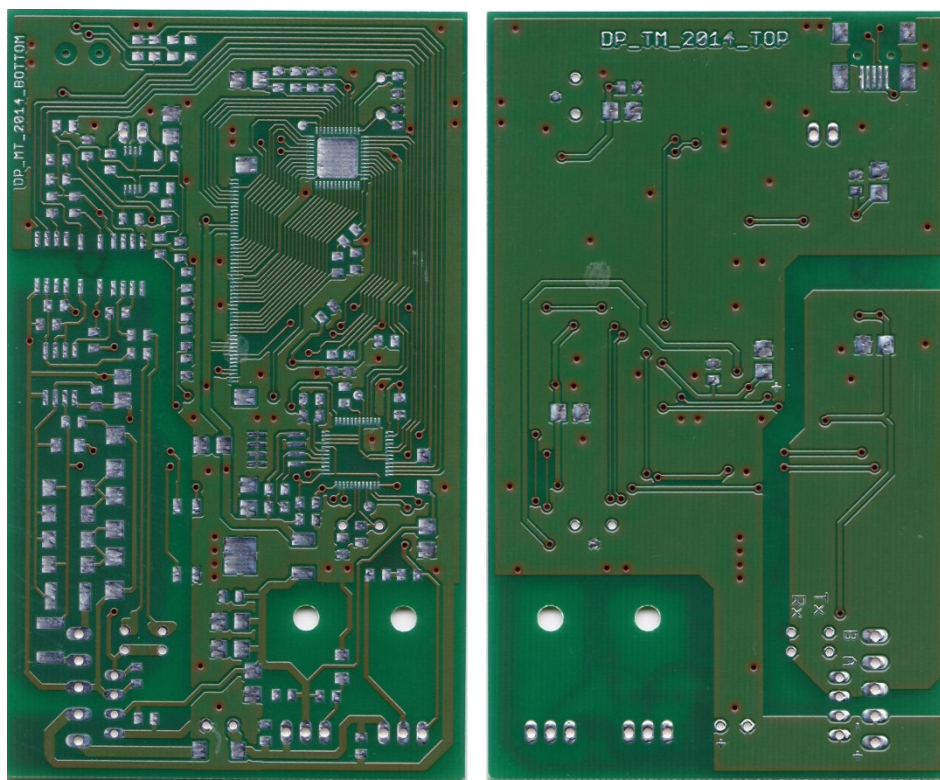
Nakonec však zapájení proběhlo bez větších problémů. Spodní ploška byla připájena pomocí otvoru, který byl dodatečně (před osazením čipu) vyvrtán do středu plošky tak, aby bylo možné hrotem pájky roztavit cín.

Datové spoje mají šířku minimálně 0,2mm. Napájení je taženo spoji s větší šířkou. Ta je úměrná proudu, který spojem prochází. Při úpravě desky na dvě vrstvy byla snaha o to, aby byla v horní vrstvě rozlita zem. Veškeré napájecí části jsou filtrovány keramickými a tantalovými kondenzátory umístěnými co nejblíže integrovaných obvodů. Tantalové kondenzátory byly použity pro svoji nízkou výšku a kvalitnější konstrukci. Jejich nižší kapacita oproti elektrolytickým hliníkovým kondenzátorům byla kompenzována větším počtem součástek a jejich plošným rozmístěním blíže ke zdrojům rušení. Prokovené otvory jsou větší, než je obvyklé pro tuto třídu přesnosti. Je to z toho důvodu, že deska je spíše testovací a bude-li třeba, může se do otvoru připojit vodič o průměru 0,6mm.

Část desky, kde je realizován převodník z TTL na RS485 je připravena tak, aby bylo možné tuto část galvanicky oddělit. Proto je tato část oddělena od zbytku desky izolační bariérou. Bariéra je nejúžší v místě DC/DC převodníku. Nemí předpoklad, že se toto oddělení použije, ale během testování se mohou objevit další požadavky, které se na vyrobené desce těžko dodatečně realizují.

Pro výrobu desky byla vybrána firma Gatema. Tato firma má bohatou tradici ve výrobě plošných spojů. Na svých webových stránkách má jednoduchý poptávkový dotazník pro určení ceny. Navíc je možná dohoda na vzorkové slevě pro studenty. Obrys desky by měl být správně vyfrézovaný tak, aby správně zapadl konektor a aby bylo možné protáhnout kabel od displeje do spodní části krabičky. Frézování by desku plošných spojů zbytečně prodražilo. Proto budou tyto výřezy upraveny ručně před osazováním. Jedná se o vzorek.

Do výroby byla odeslána data pro vrtací zařízení ve formátu excellon.cam. Pro fotoplotter byla odeslána data ve formátu gerber gerb274x.cam. Tato data vygeneroval program Eagle.



Obr. 57 Vyrobená deska plošných spojů (obě strany)

Deska byla vyrobena a doručena v termínu. Osazení desky potom probíhalo postupně od zdroje, přes procesor a potom následovaly periferie. Podrobnosti jsou popsány v kapitole 5.5.



## 5. Program

Při tvorbě nových projektů by se měla dodržet předepsaná posloupnost. Největším nepřítelem při práci bývá chaos. Čas věnovaný správnému rozplánování následující práce většinou ušetří mnoho času při realizaci. Tak jako u řady jiných činností, je nutné i u programování myslet trošku dopředu. Posloupnost při tvorbě programu je popsána v mnohých knihách a zní asi následovně:

- *zadání úkolu*
- *analýza projektu*
- *volba jazyka*
- *programování*
- *ladění aplikace*
- *testování*
- *ukončení projektu*

### 5.1. Zadání úkolu

Zadání proběhlo při zadání tématu této práce. Mělo by se navrhnout zařízení, které bude ovládat kontrolér pro krokové motorky. Přímou se má jednat o kontrolér R356 od firmy RMS Technologies Inc.

### 5.2. Analýza projektu

Z hlediska programu by mělo jít o zařízení, kdy na vstupu je zadání řetězce ASCII znaků a výstupem je odeslání tohoto řetězce na linku RS485. Program bude obsluhovat tři základní části. Dotykový panel, displej a komunikaci.



Obr. 58 Tři základní části zařízení

**Dotykový panel** je součástí displeje a bude plně nahrazovat tlačítka. Pomocí dotykového panelu se budou zadávat znaky, nastavovat rychlost a další možnosti. Dotykový panel je vyhodnocován čipem FT800, takže procesor nemusí měřit jednotlivé elektrody, jen si vyčítá naměřené hodnoty pomocí SPI komunikace. Kontrola stisknutí panelu musí probíhat cyklicky několikrát za sekundu. Ideální čas je 50 - 100ms.

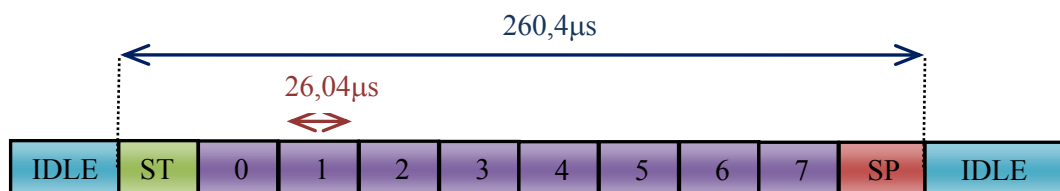
Tím se vytvořil požadavek na časovou základnu. Program musí být synchronizován časem. Není však nutné používat reálný čas. Informace o přesné hodině a minutě by se nevyužila. Tím pádem bude na časovou základnu stačit přerušení od nějakého časovače. RTC obvod zálohovaný baterií není potřebný.

Vychází z toho úvaha, že program bude tvořený hlavní nekonečnou smyčkou, která bude synchronizována každých 50-100ms a časovačem, který bude vyvolávat přerušení. Čas přerušení se odvíjí od nejkratšího času, který se obsluhuje. Zatím je to 50ms. Ale to se týká jen tlačítek, v našem případě dotykového panelu. Nesmíme zapomenout na ostatní dvě části zařízení. Displej nebude potřebovat žádnou obnovovací frekvenci, žádné cyklické přepisování dat. Obrázky se nastaví spíše jednorázově. O vše ostatní se stará čip FT800.

**Komunikace** na rozdíl od dotykového panelu vyžaduje, aby byl přijatý byte uložen dříve, než bude v přijímacím bufferu přepsán dalším. To se může dělat různými způsoby. Vyvolá se přerušení při příjmu, nastaví se DMA, anebo se cyklicky kontroluje přijímací flag. V naší aplikaci by se toto nemuselo vůbec řešit, protože příjem se očekává pouze po odeslání příkazu. Pak stačí přepnout zařízení na příjem a čekat po dobu časového limitu, zda něco nepřišlo. Takovéto zjednodušení by se

však mohlo vymstít při následném rozšiřování aplikace. Přímou se například nabízí využít zařízení pro monitorování linky RS-485, a podobně. Pak by se musel program zbytečně přepisovat.

Při uvážení, že procesor je rychlý a výkonný a aplikace je vcelku nenáročná na čas, nabízí se možnost načasovat timer tak, aby byl rychlejší než naplnění komunikačního bufferu a již v přerušení tento buffer obsloužit. K tomu je nutné zjistit, jak dlouho trvá příjem jednoho byte do bufferu. Nejvyšší rychlost je 38 400 Bd. To znamená, že jeden bit přijde za 26,0417 $\mu$ s. Nejkratší přijaté slovo bez paritního bitu a s jedním stopbitem je dlouhé deset bitů. Tím pádem se musí vykonání programu v přerušení opakovat periodicky minimálně každých 260 $\mu$ s. Jinak by mohlo dojít k přepsání bufferu jiným znakem.



Obr. 59 Minimální délka jednoho byte při rychlosti 38,4kBd

Použitý komunikační protokol je RS-485, který je poloduplexní, a proto se nesmí zapomenout po odvysílání posledního znaku uvolnit komunikační linku. Vypnout vysílací obvod a připojit přijímací část obvodu.

**Displej** vyžaduje spíše jednorázovou obsluhu. Komunikace s displejem bude sice náročná na množství přenesených dat, ale nebude vyžadovat cyklickou obsluhu. Skutečnou obsluhu displeje na sebe bere čip FT800. Program bude jen nastavovat a měnit obrazy.

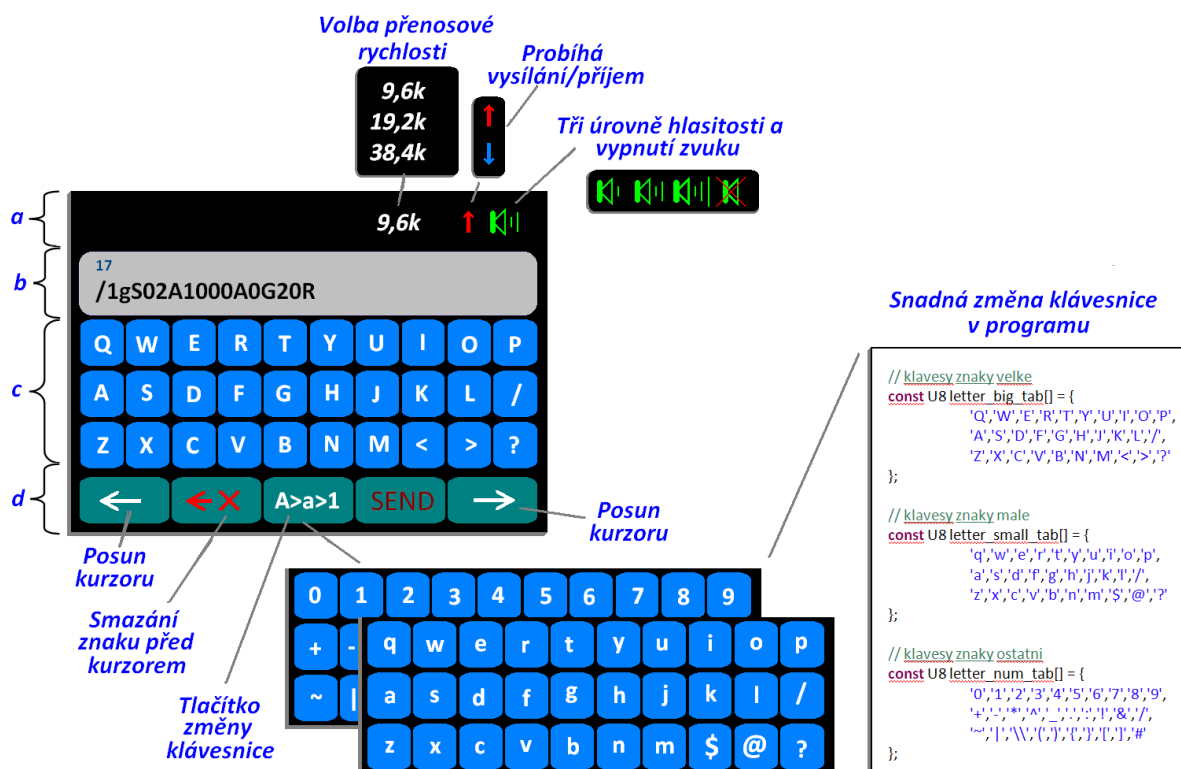
Rozlišení displeje a skutečnost, že je barevný, dává aplikaci nový rozměr. První testy s monochromatickým displejem 128x64 bodů ukázaly, že by tento displej vyhověl, ale ovládání by bylo docela náročné. S novým displejem je možné vytvořit i klávesnici, nebo nastavovací tlačítka. Omezení zde nastává u rozměru políčka vymezeného na tlačítko. Pokud se chceme vyhnout nutnosti použití tužky, která se ráda ztrácí, musíme ctít rozměr prstu. Tím se omezuje počet tlačítek zobrazených na displeji. Na obrázku Obr. 60 je návrh jak by mělo vypadat rozmístění tlačítek a textu na displeji. Tento návrh bude základ pro celou aplikaci.

Jak je vidět na obrázku, plocha displeje je rozdělena na čtyři základní sektory. Původní myšlenka byla taková, že bude použito více obrazovek. Během prvních testů s displejem se ukázalo, že je to vcelku zbytečné a matoucí. Po krátkém vnitřním boji bylo od této myšlenky upuštěno a displej byl rozdělen do již zmiňovaných sektorů.

**Sektor a** je umístěn u horního okraje displeje. Jedná se spíše o informační lištu, která obsahuje různá systémová nastavení, nebo ovládací prvky. Ty jsou dobře viditelné a dobře přístupné. Úplně vpravo je ikonka pro nastavení zvuku. Stisknutím této ikonky je možné zvolit tři úrovně hlasitosti, nebo úplně vypnutí zvuku. Nastavená úroveň se zapíše do nevolativní paměti. Vedle ikonky pro nastavení zvuku je číslo, které odpovídá přenosové rychlosti. Stisknutím tohoto čísla se mění přenosová rychlost v hodnotách 9600Bd, 19200Bd a 38400Bd. Zbytek sektoru a je zatím volný a je připraven pro další využití. Vlevo od přenosové rychlosti jsou připravena tři tlačítka.

**Sektor b** je vlastně příkazový řádek, na kterém se pohybuje kurzor. Zde se zobrazují znaky, které se následně odešlou po lince RS485. V levém horním rohu se průběžně ukazuje délka textu.

**Sektor c** tvoří klávesnice. Protože se nevešly všechny znaky do jedné klávesnice, jsou k dispozici tři klávesnice, které se postupně přepínají. První klávesnici tvoří velká písmenka, druhou malá písmenka a třetí klávesnice obsahuje čísla a ostatní znaky. Programově jsou tyto klávesnice řešeny tak, že jim odpovídají pole konstant. V těchto polích je možné znaky na klávesnici jednoduše editovat. Po odeslání se sektor c změní na monitor odeslané a přijaté zprávy.



Obr. 60 Rozložení ovládacích prvků na displeji

**Sektor d** je složen z pěti větších tlačítek. Vlevo je tlačítko pro posun kurzoru vlevo. Vedle něho je tlačítko pro mazání znaku umístěného vlevo od kurzoru. Uprostřed je tlačítko pro změnu klávesnice. Jak již bylo zmíněno, tlačítko má tři různé úrovně. Při stisku se ikonka na tlačítku změní tak aby zobrazovala to, co následuje. Vpravo od středu je tlačítko pro odeslání. Po jeho stisknutí se odešle připravený textový řetězec a jako poslední znak se vloží <CR>. To je nezobrazitelný znak, jehož ASCII hodnota je 13. Vpravo dole je potom tlačítko, kterým se posunuje kurzor doprava.

Samotný obraz je tvořen programově z obdélníků různé barvy. Následně se na displej umístí znaky nebo nápisy. Každý znak má svoji velikost a polohu. Při tvorbě obdélníku se zadá pouze pozice všech hran obdélníku (resp. čtverce). Barvy se nastavují jednotně pro více objektů najednou. Během sestavování obrazu bylo využito jednodušší varianty kreslení.

Sestavení obrazu je vcelku jednoduché, je však nutné si dopředu rozpočítat správně velikosti a umístění jednotlivých komponent. Další počítání pak nastává při rozmístění tlačítek na dotykovém panelu. Zatímco displej má rozlišení 320x240, dotykový panel má naměřené hodnoty 1024x1024. Je sice možné použít funkci integrovaného ovladače, která umí naměřené hodnoty přepočítávat, ale zatím s tímto postupem nebylo dostatek zkušeností.

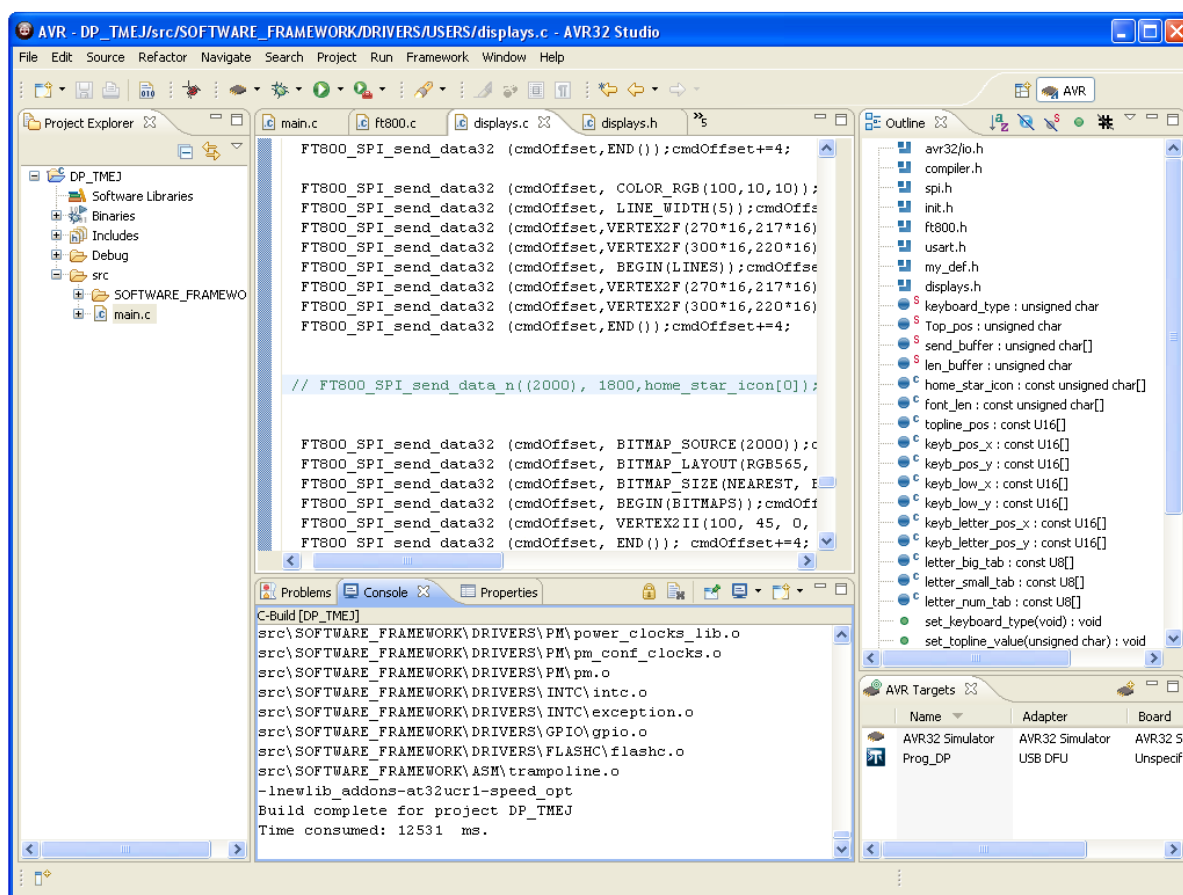
Ikonky jsou složeny z čar. Je možné umístit na displej bitmapu, ale pro použité ikonky se zdálo být použité řešení jednodušší, a také rychlejší pro přenos. Ikonka zvuku je složena z maximálně sedmi čar. Podobně byly řešeny i šipky na klávesnici.



### 5.3. Výběr programovacího jazyka a vývojového prostředí

**Výběr jazyka** byl vcelku jednoduchý. Pro programování jednočipových procesorů se užívá Assembler, jazyk C, jazyk C++, Java, někdy jazyk Pascal, jazyk Basic, potom je možné programovat v různých grafických prostředích a tak podobně. Výběr je vcelku velký. Pro vysoké využití programové paměti a pro maximální kontrolu nad procesorem je asi nejvhodnější Assembler (nebo strojový kód). Programování v Assembleru, který patří mezi takzvané nižší programovací jazyky, je však velmi pracné a náročné na čas. Také přehlednost a kontrola je o dost horší, než například u jazyka C. Svědčí o tom i skutečnost, že i zarytí programátoři Assembleru, kteří překročili bludný kořen a přešli na jazyk C, se k Assembleru neradi vrací. Současné jednočipové procesory mají dost velké paměti, a jsou dostatečně rychlé, aby nebylo použití Assembleru nutností. Vcelku rozšířený je již zmíněný programovací jazyk C. Je to tak rozšířený jazyk, že ho není nutné ani nijak zvlášť představovat. Z historie snad jen, že byl vyvinut počátkem 70. let 20. století. Jeho autoři jsou Ken Thompson a Denis Ritchie, kteří ho vyvinuli pro potřeby operačního systému Unix. Na jeho platformě potom vznikly další programovací jazyky. Základem tohoto jazyka je jen pár příkazů. Podrobnější popis je možné získat na internetu, takže se tímto nebudeme zdržovat. Podstatné je, že výrobci procesorů a některých dalších integrovaných obvodů mají na svých stránkách vzorové programy pro své výrobky, které jsou napsány právě v jazyku C. To je velká pomoc pro stavbu jakéhokoliv zařízení. Ušetří to čas a usnadní řešení problémů. Tolik k výběru programovacího jazyka.

**Výběr vývojového prostředí** je o něco horší problém. Jak již bylo zmíněno, jedním z důvodů proč byl vybrán tento procesor, byla skutečnost, že jeho výrobce na svých stránkách nabízí vývojové prostředí zdarma.



Obr. 61 Vývojové prostředí AVR32 studio [18]

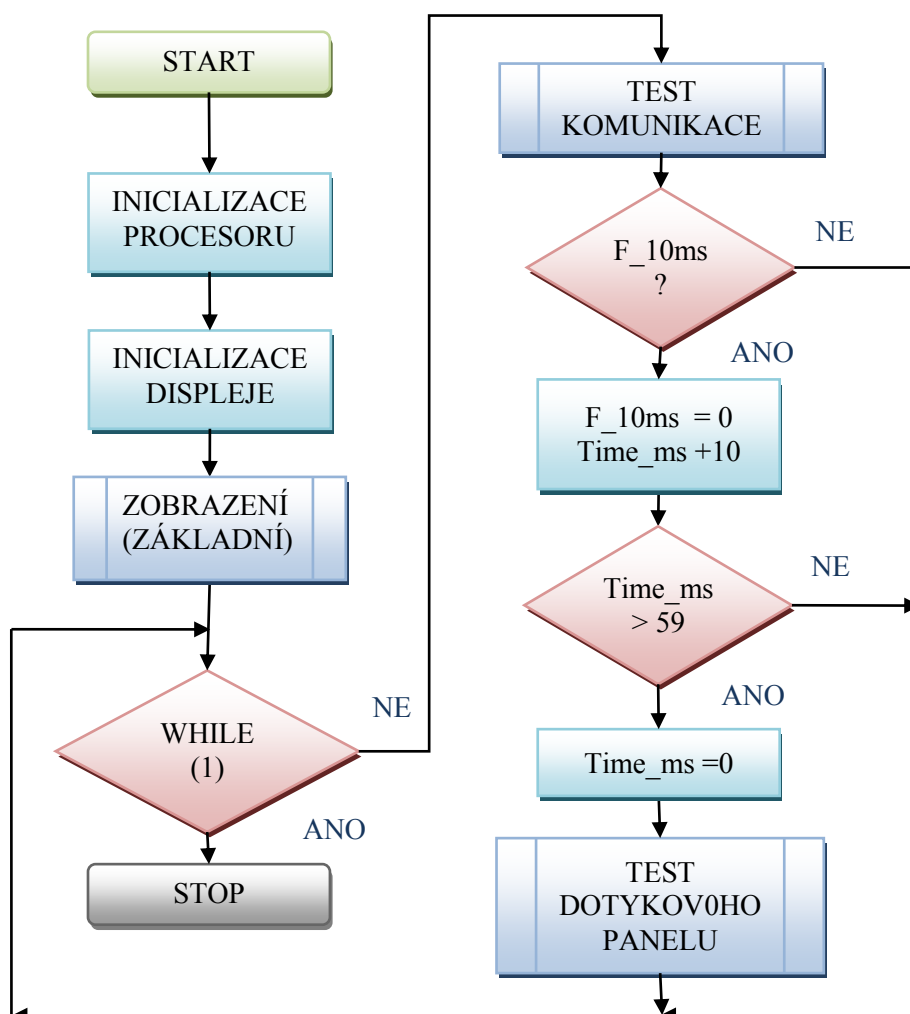
Jde o standardní prostředí, které nabízí mnoho knihoven a příklady aplikací. Takovéto prostředí poskytuje mnoho dalších výrobců, například Microchip, Freescale, Texas instrument, atd. Některé programy jsou však zpoplatněny, nebo omezeny, jinde je problém s naprogramováním čipu. Byla by nutná další investice do programátoru. V neposlední řadě jsou někdy problémy se získáním samotného procesoru. Ale to již bylo zmiňováno při volbě procesoru. Každopádně jsou knihovny a vzorové aplikace velkým pomocníkem a velmi urychlují práci.

Firma ATMEL nabízí, již mnoho let, pro svoje jednočipové procesory jednotné prostředí ATMEL STUDIO. Toto vývojové prostředí je možné stáhnout na stránkách výrobce. Oplátkou za tento program je vyžadována registrace. Když už jsme na těchto stránkách, je dobré si stáhnout prográmek FLIP, který později pomůže s programováním samotného čipu.

Při instalaci ATMEL STUDIA jsem měl problém s Windows XP. Možná i zastaralý hardware mého počítače byl příčinou, že toto prostředí nefungovalo spolehlivě. Proto jsem zvolil starší prostředí AVR32 STUDIO. Není již výrobcem podporováno, ale fungovalo spolehlivě a vytvořené programy je možné snadno importovat do nového prostředí.

#### 5.4. Psaní programu

V kapitole 5. 2. jsme si určili základní rysy programu, kterých se budeme držet. Vycházíme z toho, že program obsahuje jednu hlavní „nekonečnou“ smyčku a jedno přerušení. Hlavní smyčka se opakuje v čase 60ms. Přerušení je vyvoláno přetečením timeru a opakuje se po 200ms.



Obr. 62 Vývojový diagram pro hlavní část programu

### 5.4.1. Popis hlavní části programu

**Start** je symbol pro začátek programu. Program se začíná vykonávat po připojení napětí a rozběhu oscilátoru. Může začít i po resetu. Reset je vyvolán buď externě přes pin, ten je však v našem případě nevyužit. Nebo při podpětí, kdy začne fungovat obvod zvaný „brown out detector“. Tento detektor by měl zabránit bloudění programu při podpětí, nebo velkém rušení s podpětíovými špičkami, kdy se vyskytují náhodné stavy. Reset může vyvolat i takzvaný „watchdog timer“, který běží nezávisle na hardware procesoru. Pokud není včas vynulován, vyvolá reset. Mělo by se tím opět zabránit bloudění nebo zamrznutí procesoru.

**Inicializace procesoru.** Je to vlastně základní uživatelské nastavení všech vlastností, procesoru. Zde jsou využity knihovny „avr32/io.h“, „tc.h“, „power\_clock\_lib.h“ a „usart.h“, které poskytuje ke svému procesoru firma Atmel. Jedná se především o nastavení oscilátoru, potažmo PLL smyčky, timeru s pořadovým číslem 0 a asynchronního sériového portu. Zároveň se nastavují příslušné vstupní a výstupní piny procesoru. Všechny rutiny, které se týkají inicializace jsou v souboru s názvem „init.c“.

**Inicializace displeje** nastává hned po nastavení procesoru. Pro upřesnění. Nejedná se o displej, ale o čip FT800. Ale tento čip se nastavuje podle typu displeje k němu připojeného. Takže je to jakýsi celek. Při startu se musí čtít určitá prodleva, protože se čip musí nejprve probudit. Chvilku trvá, než začne komunikovat. Tak jako procesor, má i integrovaný obvod FT800 od výrobce bohatou podporu. Sice nebylo možné využít knihovny přímo, protože jsou psány buď pro PC, nebo pro jiný typ procesoru, ale nedalo takovou práci kód upravit. Při sestavování této části programu nebylo co vymýšlet, bylo nutné se držet postupu. Pro všechny rutiny použité pro inicializaci a komunikaci (SPI) s displejem byl vytvořen soubor s názvem „FT800.c“.

**Zobrazení** je podprogram, který vykreslí celý displej. Po připojení napětí je vhodné co nejdříve displej rozsvítit a nezdržovat uživatele zbytečnými obrázky s logem. Podprogram „zobrazení“ je využit i v dalších částech programu a podle vstupních parametrů je možné volit typ obrázku. Podprogram „zobrazení“ je v souboru „display.c“.

**Hlavní smyčka**, kterou tvoří příkaz „while“ s takovou podmínkou, která je vždy splněna. To zaručí, že program bude v této smyčce obíhat až do odpojení napájení. Pokud nepřijde nějaký reset, nebo jiný problém. V úvodu jsem trochu mylně uvedl, že tento cyklus probíhá jednou za 60ms, ale to není pravda, protože cyklus probíhá nezávisle. Některé podprogramy jsou však synchronizovány na čas 60ms. Při tvorbě programu bylo dbáno na to, aby bylo dostatek času na to. Aby cyklus alespoň jednou za 60ms proběhl.

**Test komunikace** je další podprogram, který je v hlavní smyčce. Stará se o komunikaci na lince RS485. Volání programu leží mimo podmínky na čas 60ms i mimo podmínky na čas 1ms, proto je vykonáván velmi často. Vykonání tohoto podprogramu zabere maximálně 20μs (při 12MHz), a to jen v případě přijaté zprávy. Ve většině případů se pouze testuje jedna nebo dvě podmínky, což zdržuje běh programu jen minimálně.

**Podmínka deseti milisekund** funguje tak, že program v přerušení nastavuje každou desátou milisekundu příznak, který je sdílený s hlavní smyčkou. Deset milisekund bylo vybráno z toho důvodu, že žádný podprogram netrvá déle a zároveň je to dostatečně jemný čas na to aby bylo možné v budoucnu provádět další úkony. Toto časování je vytvořeno tak, aby při náhodném překročení deseti milisekund nedošlo ke zkolabování programu. Nejhorší věc, která tímto nastane, bude zpomalení odezvy na stisknutí dotykového panelu. Jestliže proběhne čas 10ms a v přerušení se nastaví příznak, je vyhodnocena podmínka, příznak se zruší a přičte se hodnota deset k jednobajtové proměnné Time\_ms. Tato proměnná je vytvořena proto, aby bylo možné vytvořit smyčku s opakovacím časem 60ms.

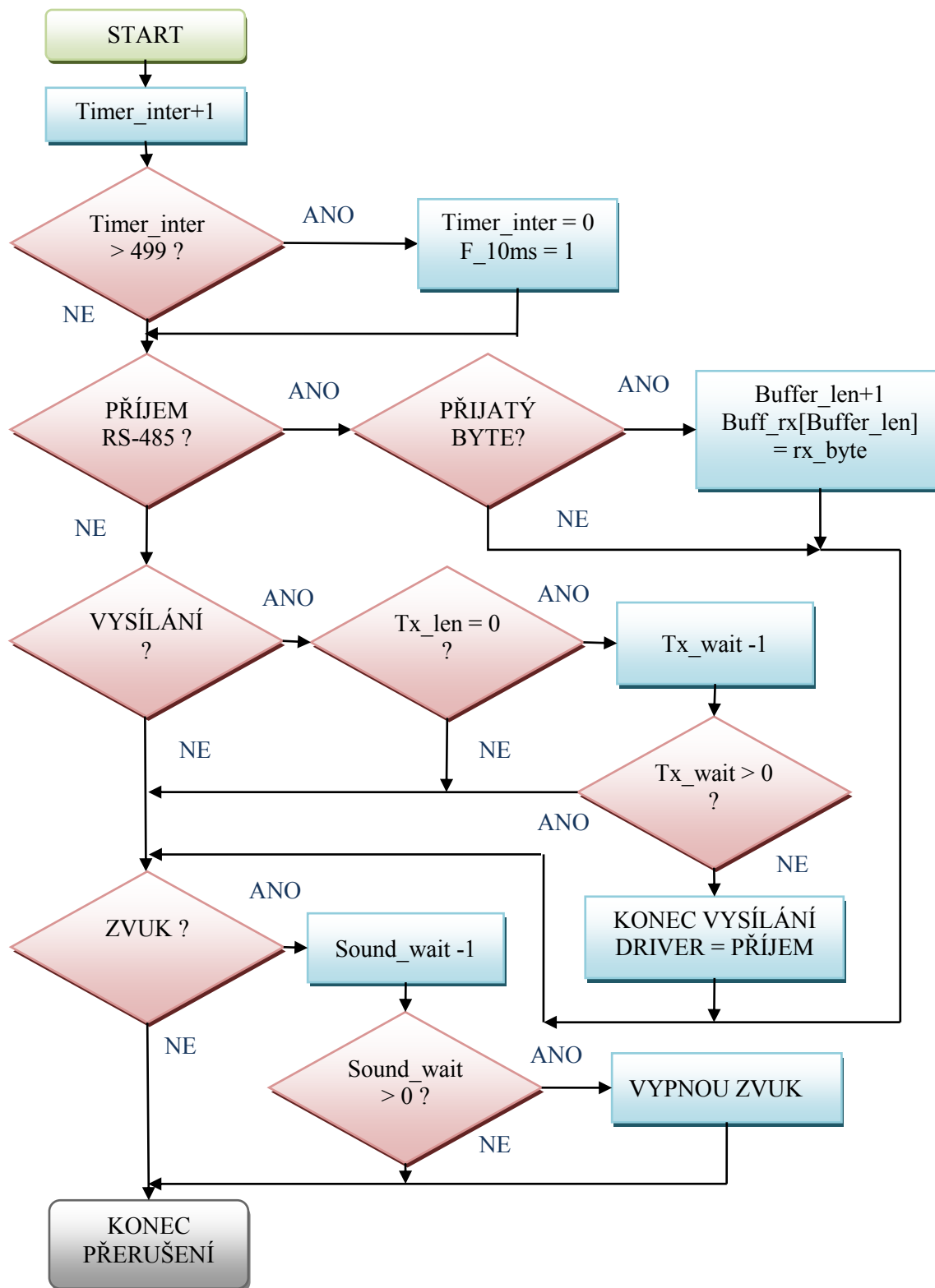
**Podmínka šedesáti milisekund** je vyhodnocena z hodnoty proměnná Time\_ms. Pokud je hodnota této proměnné vyšší než padesát devět, proměnná Time\_ms se vynuluje a provede se kontrola dotykového panelu.

**Test dotykového panelu** obsahuje mnoho dalších podprogramů a bude dále podrobněji rozebrán. Jeho hlavním úkolem je vyčíst data z integrovaného obvodu FT800 a vyhodnotit zda je stisknutý dotykový panel. Pokud je stisknutý, pak se zjišťuje pozice stisknutí a tak dále.

Po ukončení podprogramu se procesor vrací k základní podmínce „WHILE“, a celý cyklus se opakuje.

#### 5.4.2. Popis programu vykonaného při přerušení

Program v přerušení by měl být co nejkratší, protože se pakuje v krátké periodě a hlavní program běží v mezerách mezi přerušením. Čím budou mezery kratší, tím více se bude hlavní program zpomalovat.



Obr. 63 Vývojový diagram pro program vykonávaný v přerušení

Vývojový diagram znázorňuje program, který se aktivuje při přerušení od časovače (timeru) s pořadovým číslem nula. Tento podprogram má na starosti čtyři základní věci.

1. Nastavovat příznak každých 10ms. Tento příznak je pak dále využíván na synchronizaci hlavní smyčky.
2. Kontrolovat přijímací registr. To je zvlášť důležité, protože při pozdním výběru tohoto registru může dojít ke ztrátě přijatých dat. Každý přijatý byte se ukládá do zásobníku a zároveň se zvyšuje počítadlo přijatých dat Rx\_len. Před inkrementací tohoto počítadla se kontroluje maximální délka zásobníku (bufferu), aby tento zásobník nepřetekl.
3. Přerušení je také využíváno ke kontrole toku dat na lince RS-485. Před vysíláním se uvede driver do stavu „vysílání“. Tento stav by však po dokončení vysílání blokoval linku a nebylo by možné přijmout žádnou odpověď. Proto se po odvysílání posledního znaku čeká cca 4 ms a driver se přepíná na příjem. Protože nejrychlejší synchronní časování s rozlišením 200 $\mu$ s je přímo v přerušení, je zde část programu, která se o přepnutí driveru postará.
4. Poslední část byla přidána jaksi navíc. Pokud by se pro tuto aplikaci použila tlačítka, byl by možné po hmatu a z cvaknutí zjistit, že tlačítko bylo stisknuté. U dotykového panelu není žádná takováto zpětná vazba. Proto se využívá zvukového výstupu. Například pípnutí, cvaknutí a podobně. Při ladění programu byl slyšet šum v době, kdy byl zesilovač zapnutý a nebyl vyslán žádný tón. Proto a také z důvodu nižšího odběru je zesilovač po doznění tónu vypínán. K tomu je v přerušení časovač „Sound\_wait“, který se před zapnutím tónu nastaví a při vynulování vypíná zesilovač.

Během ladění bude čas přerušení změřen osciloskopem, a samotný program upraven tak, aby zdržení bylo co nejkratší.

### 5.4.3. Hlavní podprogramy

#### 5.4.3.1. *Test komunikace*

Podprogram, který testuje komunikaci, se zajímá vlastně jen o přijaté zprávy. Zpráva, kterou je nutné vyslat je dlouhá maximálně 256 bajtů, což je při nejnižší rychlosti prodleva maximálně 270ms. Pokud se přidá ještě 60ms mezi dotyky, vyjde celková prodleva 330ms. Tento čas je však výjimečný, neboť délka většiny zpráv je pouze pár bajtů. Na tuto dobu je možné přestat kontrolovat dotykový panel a vyčkat na odeslání celé zprávy. Vysílání navíc probíhá ihned po stisku „tlačítka send“. Další stisk, v čase kratším než 500ms, by byl brán jako zákmit nebo porucha. Vysílání se tímto zjednodušilo.

Přijaté znaky však mohou přijít kdykoliv. A protože se v přerušení plní zásobník, je tento zásobník nutné někdy vybrat a vyčistit. V režimu, kdy se znaky zobrazují on-line na displej je potom každá prodleva zbytečná. Proto je tento podprogram umístěn v pozici, kdy může být vykonáván rychle.

#### 5.4.3.2. *Test dotykového panelu*

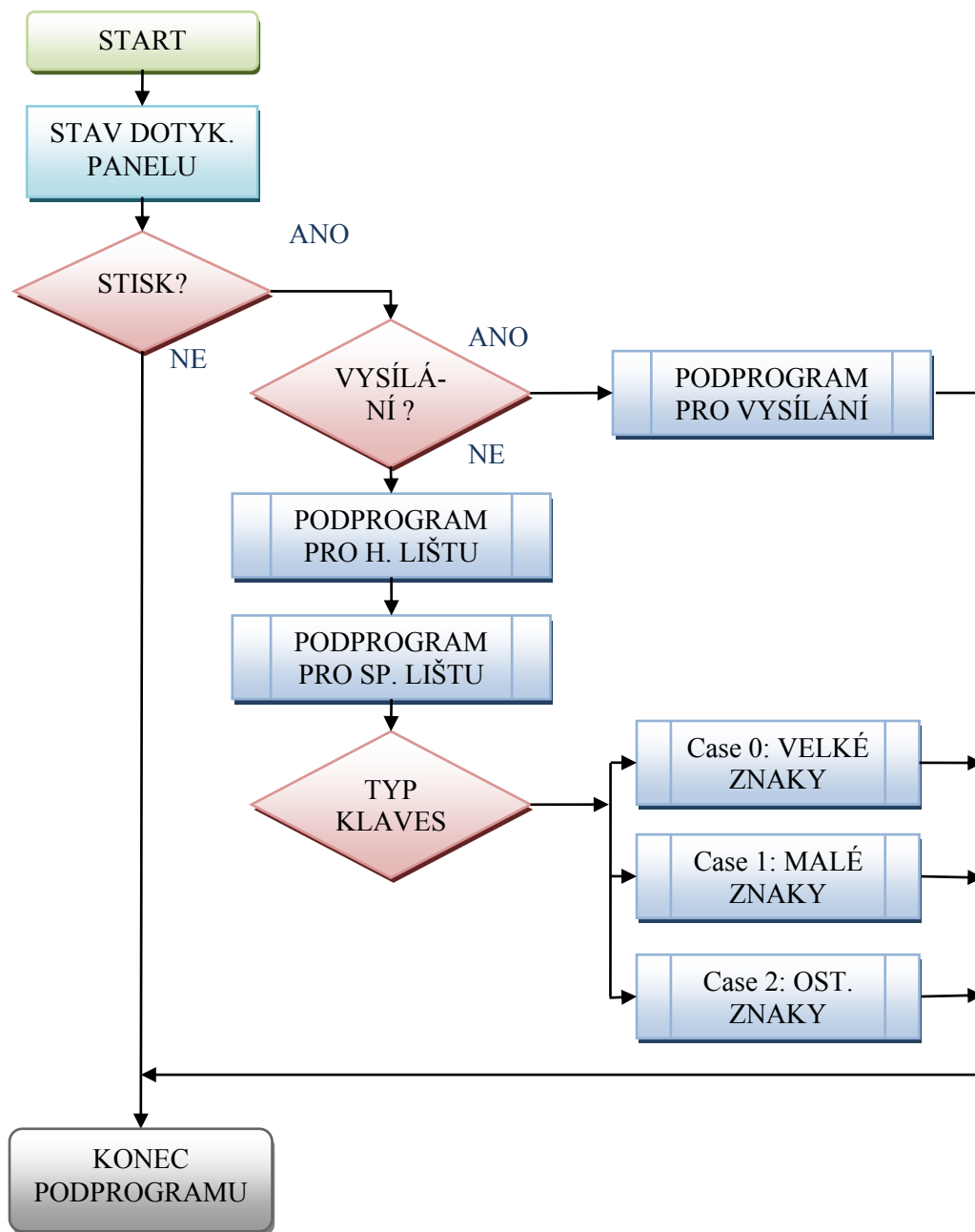
Hlavní část celého programu se skrývá v tomto podprogramu. Tento podprogram nejen testuje stisk a pozici stisku dotykového panelu, ale stará se i o vykonání funkce příslušného tlačítka, což je skoro vždy spojeno s překreslením displeje. Překreslení displeje vypadá jako složitá záležitost, protože 320x240 bodů je 7200 a k tomu si musíme uvědomit, že každý znak má tři barvy v rozlišení 6 bitů.

Tady se projevuje, jak moc zjednodušuje FT800 ovládání displeje. Například nakreslení celé klávesnice je otázkou pár řádků. Rychlost překreslení obrazu potom závisí spíše od rychlosti SPI portu. Vývojový diagram zachycuje tuto část programu jen v hlavních rysech. Detaily by se do zobrazení nevešly. Celý program je přiložen k této práci.

Podprogram je volán periodicky a ihned po jeho startu se přečte stav dotykového panelu. Stav se zjistí přečtením příslušných registrů z integrovaného obvodu FT800. Integrovaný obvod měří odpory dotykového panelu v osách X a Y. Ty potom ukládá do více registrů v různých formátech.

V našem případě vyčítáme pouze dvě desetibitová čísla. Pokud je naměřená hodnota menší než 1023, je dotykový panel stisknut. Správným rozdělením panelu je možné identifikovat polohu stisku. Tímto způsobem se pak kontrolují jednotlivá tlačítka.

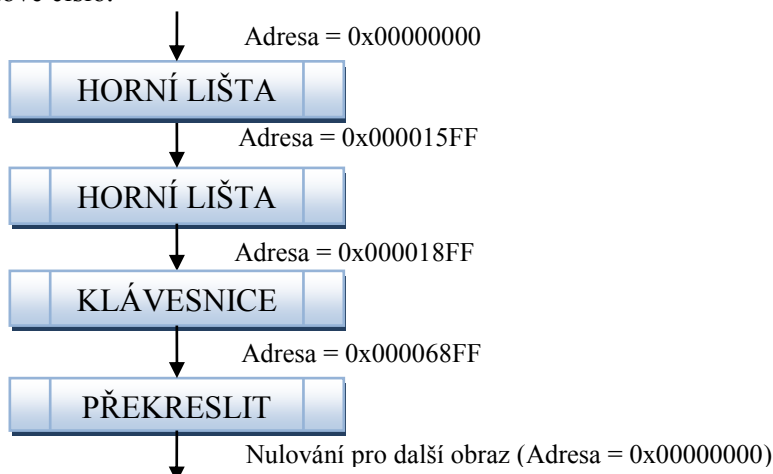
Plocha displeje je rozdělena na pevně tak aby odpovídala jednotlivým tlačítkům. Rozložení vychází z Obr. 60. Toto rozložení je používáno v celém programu mimo podprogramu pro vysílání. Tam se snímá pouze tlačítko „OK“ Obr. 70.



Obr. 64 Podprogram „Test dotykového panelu“

#### 5.4.3.3. Podprogramy pro obsluhu displeje

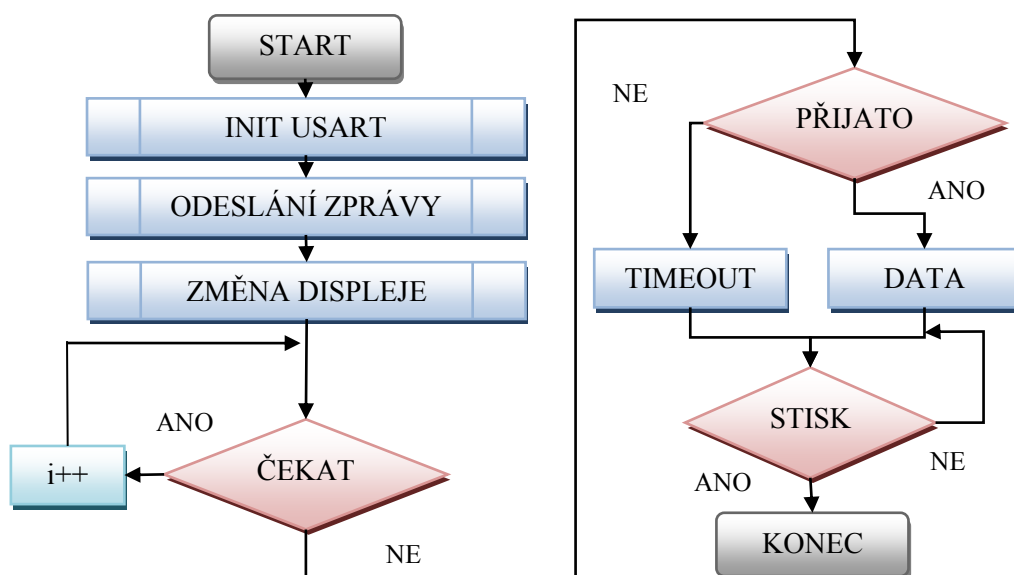
Veškeré podprogramy pro obsluhu displeje jsou sestaveny tak, aby odesílaly, pomocí SPI komunikace jednotlivé příkazy do paměti čipu FT800. Během vytváření obrazu se musí adresa paměti inkrementovat a odesílat před každým příkazem tak, aby se příkazy skládaly do paměti jeden za druhým. Jakmile je poskládán celý obraz, je dán příkaz k překreslení a zároveň se nuluje výchozí adresa. Všechny rutiny, které se starají o skládání obrazu, mají jako vstup i výstup hodnotu adresy. Například po vykreslení horní lišty se vrátí hodnota, která se posílá do další rutiny vykreslující klávesnici. Tato adresa je 32-bitové číslo.



Obr. 65 Konstrukce obrazu a jeho odeslání na displej

#### 5.4.3.4. Podprogramy pro vysílání

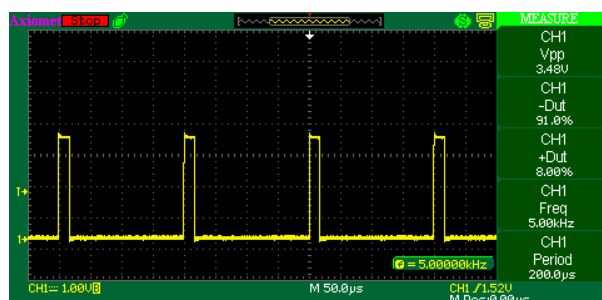
Pro vysílání a příjem odpovědi používá program zvláštní podprogram. Tady se nejprve nastaví na sériový port správná rychlost, driver se přepne na vysílání a odešle se zpráva s přidaným znakem <CR> na konci. Přerušení se potom postará o přepnutí driveru zpět na příjem. Následně se vynulují přijímací registry a program čeká 800ms na příjem zprávy. Během tohoto času se displej překreslí na monitorování komunikace Obr. 70. Pokud odpověď nepřijde, vypíše se na displej zpráva „TIMEOUT“. Pokud přijde, zobrazí se přijaté znaky. Nezobrazitelné znaky se nahradí tečkou. Přijatá odpověď se částečně dekóduje tak aby bylo jasné, zda zpráva dorazila v pořádku. Následně program čeká na stisknutí jediného tlačítka, aby se mohl vrátit do hlavní smyčky.



Obr. 66 Vývojový diagram podprogramu pro vysílání

## 5.5. Ladění aplikace a oživení desky plošného spoje

Část programu týkající se displeje byla již dopředu odladěna na vývojovém kitu. Tím pádem se nemusely řešit případné závady na desce a také návrh budoucí desky vycházel ze zapojení vývojového kitu.

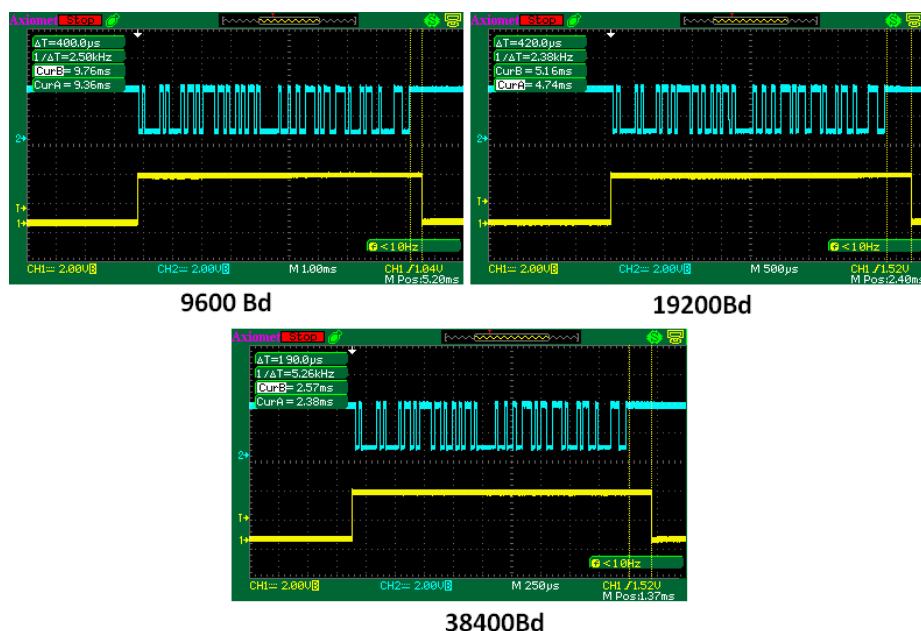


Obr. 67 Test přerušení procesoru

Takto byla odladěna inicializace displeje, obsluha dotykového panelu, zvuková část obvodu a samotná obsluha displeje. Vytvořené části programu pak byly uloženy do dvou souborů, které slouží pro obsluhu displeje. Jedná se o soubory „displays.c“, „FT800.c“ a „t\_screen.c“. Některé části programu musely být ponechány na dobu, až bude hotová finální deska. Týkalo se to hlavně komunikace RS485 a obsluha externí paměti.

Ladění programu probíhalo zároveň s ožíváním desky. Po částečném osazení plošného spoje byla deska připojena na zdroj s proudovým omezením. Ihned po připojení bylo zkontrolováno napětí na všech hlavních částech obvodu. Hlavně 5V, napětí na procesoru 3,3V a napětí jádra procesoru 1,8V. Současně byl měřen odběr. V průběhu práce bylo kontrolováno oteplení jednotlivých součástek. Kontrola probíhala jednak dotykem ruky, ale i měřením pomocí termočlánu.

Dále byla testována funkce programování. Když toto všechno proběhlo v pořádku, začal se tvořit a zároveň testovat program.



Obr. 68 Kontrola přepnutí převodníku RS485 po odvysílání při různých rychlostech

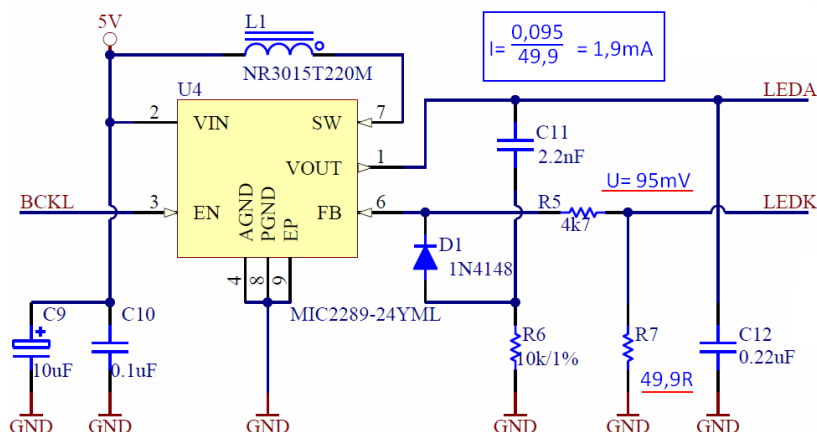
Jako první byla napsána inicializace procesoru. Součástí této inicializace je nastavení **přerušení**. Pokud bude přerušení fungovat správně, pak je správně nastaven i oscilátor. Pro ověření funkce přerušení je na začátku nastaven volný pin PA02 do jedničky a těsně před ukončením přerušení



Další část, která se oživila a odladila, byla **komunikace RS485**. Pokud tato komunikace správně funguje, je možné posílat zprávy o činnosti programu. Tyto se pak zobrazují v některém programu v počítači. Například v programu hyperterminál. Osciloskopem byla kontrolována správná funkce převodníku a také přepnutí převodníku na příjem po odvysílání zprávy *Obr. 68*.

Horší to bylo s podsvětlením displeje. Tady jsem udělal při návrhu fatální chybu. Navržený proudový zdroj fungoval sice při testování s proměnným odporem v zátěži dobře, avšak po připojení displeje se žádné světlo neobjevilo.

Řešení bylo několik. Vytvoření nové desky by ohrozilo termín odevzdání. Minimální vstupní napětí 17V by zase omezilo uživatele. Stejně by to bylo s použitím samostatného zdroje. Nakonec byl použit malý samostatný modulek se spínaným zdrojem pro LED podsvětlení. Schéma je na *Obr. 69*.



Bohužel napájení modulku je bráno ze zdroje 5V, který je již tak dost zatížen veškerou elektronikou. Původní záměr oddělit podsvětlení se nepovedl. Aby se stabilizátor co nejvíce odlehčil, snížil se proud do podsvětlení na co nejnižší mez. Při proudu kolem 1,5mA vykazoval displej dostatek jasů na to, aby byl čitelný. Konečná hodnota proudu byla nastavena na 1,9mA. Modulek byl na pevně připájen k zemnicí ploše desky a připojen třemi krátkými drátky do obvodu podsvětlení.

Stabilizátor se tím pádem dost zahřívá. Informativní měření termočlánekem mezi chladičem a integrovaným obvodem je v tabulce

Tab. 13. Vyšší napětí než 24V nebylo s ohledem na ohrožení funkčnosti výrobku testováno. Tímto se omezuje použití výrobku pro napájecí napětí mezi 12 až 24V.

Tab. 13 Informativní měření teploty mezi chladičem a integrovaným obvodem (ustálení 20min.)

Napětí [V]	Proud klidový [mA]	Průměrný proud při aktivitě [mA]	Teplota v klidu [°C]	Teplota aktivní [°C]
12	56mA	96mA	39	48
24	56mA	92mA	65	77

Aktivitou, která je uvedena v tabulce se rozumí rychlé mačkání některé klávesy s nastavením plné hlasitosti a vysíláním na linku RS-485.

Ladění dále probíhalo již jen v oblasti programu. Program byl napsán a nyní laděn tak, aby se výsledek co nejvíce přiblížil představě znázorněné na Obr. 60.

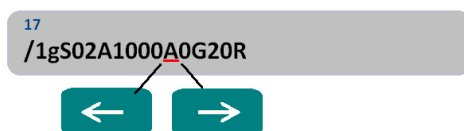
Pro odvysílání nastaveného řetězce bylo zobrazení upraveno. Myšlenka byla taková, že by měla být vidět případná odpověď, kterou zařízení potvrzuje přijetí příkazu. Pokud odpověď nepříjde do cca 1s, zobrazí se místo odpovědi nápis TIMEOUT. Příjem a vysílání zprávy také naznačuje šipka v pravém horním rohu. Šipka však neblíká současně při vysílání a příjmu. Její zobrazení by bylo velmi krátké. Po přijetí zprávy zůstane rozsvícená až do stisknutí tlačítka OK.

Z přijaté zprávy je dešifrován význam třetího přijatého znaku. Nezobrazitelné znaky jsou nahrazeny tečkou. Program čeká na potvrzení tlačítkem OK. To poskytuje obsluze dostatek času pro orientaci v odpovědi.



Obr. 70 Zobrazení odeslaných a přijatých dat

Ladění hlavní části programu pak probíhalo bez větších komplikací. Vcelku složité bylo vytvořit logiku posunu kurzoru tak aby byl text kolem kurzoru vždy správně zobrazen v příkazovém řádku.



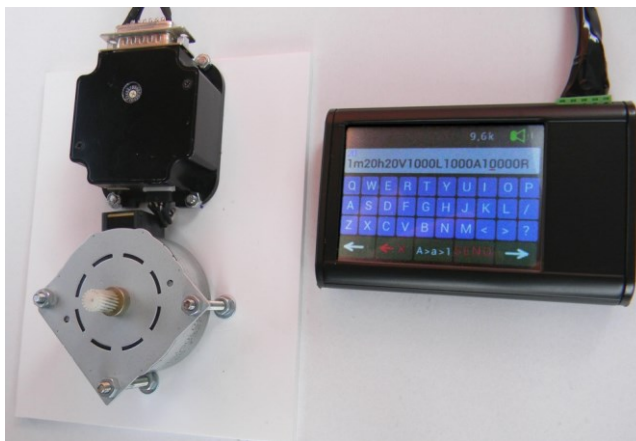
Obr. 71 Ovládání kurzoru

V příkazovém řádku může být až 254 znaků. Aby se výřez textu správně spočítal, musí se kontrolovat nejen aktuální délka textu, ale i pozici kurzoru. Použití posuvného kurzoru sice komplikuje program, ale na druhou stranu velmi usnadňuje zadávání textu. Obsluha nemusí mazat celý řádek, když potřebuje opravit chybu v zadávání. Tento typ zadávání je v současné době velmi zažitý a jeho absence výrazně ztěžuje obsluhu.

## 5.6. Testování aplikace

Testování aplikace probíhalo na zapůjčeném kontroléru R356 s připojeným dvoufázovým krokovým motorkem. Při komunikaci byl zapojen na linku RS485 převodník RS485/232 tak, aby bylo možné kontrolovat odeslané zprávy pomocí počítače. Během této kontroly se kromě hyperterminálu osvědčil také program TERMINAL V1.9b.

Vyslané zprávy byly porovnávány jednak s dokumentací firmy Lin Engineering, jednak s programem LinControl, který je určen pro ovládání a nastavování kontroléru R356. Testy dopadly dobře.



Obr. 72 Testování zařízení

## 5.7. Parametry výrobku

<b>Vnější rozměry</b>	délka, šířka, hloubka	101x72 x19 mm
<b>Materiál krabičky</b>	dural	
<b>Displej</b>	rozlišení	320x240 bodů
	úhlopříčka	3,5"
	typ	TFT
	Počet barev	3 x 6 bitů
<b>Dotykový panel</b>	typ	Odporový
		Součástí displeje
<b>Počet konektorů 2</b>	USB mini	Určený pro změnu programu
	5 pinů/3,5mm	Pin1 - RS485 / B
		Pin2 - RS485 / A
		Pin3 - ZEM
		Pin4 - ZEM
		Pin5 - +12 V až 24V
<b>Napájení</b>	stejnosměrné napětí	12V až 24V
<b>Odběr</b>	v klidu	56mA
	při stisku tlačítka a vysílání	90mA (nejvyšší úroveň hlasitosti)
<b>Odeslaná zpráva</b>	ASCII	
	max. počet znaků ve zprávě	256
<b>Sběrnice RS-485</b>	rychlost volitelná	9600Bd, 19200Bd a 38400Bd
	Počet bitů	8 bitů, 1stopbit, parita žádná
<b>Přepínače</b>	přepínač 1	ON – programování po připojení napětí
<b>(na spodní straně)</b>	přepínač 2	ON – ukonč. odpor 150R pro RS 485

## 6. Závěr

Hlavním úkolem této diplomové práce bylo vytvoření zařízení, které by mohlo pomoci s programováním inteligentních ovladačů krokových motorků. Tyto ovladače se začínají stále více prosazovat v aplikacích s krokovými motorky.

V teoretické části práce je popsáno rozdělení krokových motorků, principy na kterých pracují, způsoby řízení, principy zvyšování základních kroků a tak dále. Je zde také zmíněno jakým způsobem je možné elektronicky rozdělit základní krok krokového motoru nebo vylepšit momentovou charakteristiku. Je zde také uvedeno využití dvoufázových krokových motorků, jejich různá zapojení a způsoby řízení těchto asi dnes nejvíce používaných typů.

Další část práce se zabývá využitím průmyslových komunikací pro ovládání řídicích jednotek krokových motorů. Tento trend se začíná pomalu prosazovat do běžného provozu. Krokové motorky už nejsou řízeny výhradně pomocí pulsů určujících počet kroků a směr, ale stále více se zde uplatňuje sběrnice CAN nebo RS485. Tyto způsoby komunikace jsou v práci popsány.

Teoretická část by měla pomoci při pochopení funkce řídicího kontroléru pro krokové motorky. Nastavení a ovládání kontroléru je složité, pokud nejsou známy základní principy řízení krokových motorků.

Praktická část této diplomové práce se zabývá návrhem konstrukce a programu navrhovaného zařízení. Jsou zde postupně popsány jednotlivé komponenty a zároveň uvedeny důvody pro jejich použití. Byla zde také stanovena základní koncepce, podle které se při návrhu postupovalo.

Další kapitoly se týkají návrhu zapojení, návrhu plošného spoje a tvorbou samotného programu. Bohužel nebylo možné vypracovat podrobný vývojový diagram, proto jsou hlavní části programu zobrazeny alespoň v hlavních rysech. Celý program je potom v příloze.

Během oživování a ladění programu se vyskytlo několik problémů, které se však podařilo více-méně vyřešit. Nejzávažnějším problémem bylo podcenění návrhu zdroje. Zdroj se navrhoval jako poslední podle jednak vypočítaných a jednak částečně změřených odběrů a nebylo mu věnováno dostatek času. Správně mělo být použito spínaného zdroje jak pro hlavní napájení, tak i pro podsvětlení displeje.

Dalším problémem byla volba krabičky. Krabička byla původně určena pro jiný displej. Pro použitý displej se musela upravit.

Co se týká funkčnosti, hlavní zadání zařízení splňuje, ale nabízí se další možnosti jak ho ještě více rozšířit a vylepšit. Zvláště během testování přicházely další nápady. Některé se podařilo uskutečnit, ale ostatní bylo nutné z časových důvodů zamítnout.

Pokud by se hodnotily body, které byly dány při prvním rozboru zadání, dopadlo by to asi následovně:

1. Malé, kompaktní rozměry - splněno
2. Zařízení musí umět komunikovat po RS485 - splněno
3. Zařízení musí být schopné vytvořit a odeslat krátký řetězec složený z ASCII znaků  
Splněno. Zařízení umí složit a odeslat řetězec o délce maximálně 256 znaků, což je stejně dlouhý řetězec, jako řetězec, který umí driver R356 přijmout.
4. Komunikační rychlost 9600 nebo volitelná  
Splněno. Zařízení podporuje stejné tři rychlosti jako driver R356.
5. Napájecí napětí nejlépe v rozsahu 12V až 40V stejnosměrných  
Nesplněno, napájecí napětí je možné použít pouze v rozsahu 12-24V. Všechny vstupní součástky jsou sice dimenzovány na minimální napětí 50V, ale zdroj se při napětí nad 24V výrazněji zahřívá. I tak by měl tento snížený rozsah na běžnou práci stačit a při nutnosti vyššího napětí je možné použít malý externí zdroj.
6. Snadná obsluha – bylo velkou snahou obsluhu co nejvíce zjednodušit, avšak splnění tohoto bodu bude potvrzeno až po delších praktických testech.
7. Možnost snadné pozdější editace programu  
Splněno vyvedením USB konektoru na boku zařízení. Tímto konektorem je možné po přepnutí DIP přepínače na spodní straně zařízení přehrát program bez rozebrání zařízení.

Přínos této práce se projeví, až pokud se bude zařízení využívat ke svému účelu. To jsem se snažil ovlivnit konstrukcí a ovládáním přístroje tak, aby se jevil jako pomocník a ne jako přítěž.

Přínos této práce pro mě osobně je velký. Nejen, že jsem se blíže seznámil s krokovými motory, ale využil jsem i nové součástky. Ne vše se sice povedlo podle očekávání, ale nakonec byl vytvořen funkční vzorek.

## Bibliografie

- [1] V. V. Athani, v *Stepper Motors : Fundamentals, Applications And Design*, New Delhi, New Age International, 1997.
- [2] Google, „Electronically controlled synchronous electromotor,“ Google, [Online]. Available: <http://www.google.com/patents/US2750548>. [Přístup získán 1 4 2014].
- [3] P. NOVÁK, v *Mobilní roboty: pohony, senzory, řízení. 1. vydání*, Praha, Ben-technická literatura, 2005, pp. 41-69.
- [4] G. Lazaridis, „How Stepper Motors Work,“ 7. 5. 2010. [Online]. Available: [http://www.pcbheaven.com/wikipages/How\\_Stepper\\_Motors\\_Work/?p=1](http://www.pcbheaven.com/wikipages/How_Stepper_Motors_Work/?p=1). [Přístup získán 5. 11. 2013].
- [5] „Stepper motors,“ [Online]. Available: [http://www.allaboutcircuits.com/vol\\_2/chpt\\_13/5.html](http://www.allaboutcircuits.com/vol_2/chpt_13/5.html). [Přístup získán 12. 2. 2014].
- [6] „NEMA Motor,“ [Online]. Available: [http://reprap.org/wiki/NEMA\\_Motor](http://reprap.org/wiki/NEMA_Motor). [Přístup získán 12. 3. 2014].
- [7] „Lin Engineering,“ [Online]. Available: <http://www.linengineering.com/resources/3dmodels.aspx>. [Přístup získán 7. 2. 2014].
- [8] [Online]. Available: <http://www.parkermotion.com/catalog/cataloga/sectiona.pdf>. [Přístup získán 5. 4. 2014].
- [9] „The Variable Reluctance Stepper Motor,“ [Online]. Available: <http://www.wisc-online.com/objects/ViewObject.aspx?ID=IAU14208>. [Přístup získán 28. 2. 2014].
- [10] „AN2974,“ 6 2005. [Online]. Available: [http://www.freescale.com/files/microcontrollers/doc/app\\_note/AN2974.pdf](http://www.freescale.com/files/microcontrollers/doc/app_note/AN2974.pdf). [Přístup získán 7. 3. 2014].
- [11] „Unipolar vs. Bipolar,“ [Online]. Available: [http://probotix.com/stepper\\_motors/unipolar\\_bipolar/](http://probotix.com/stepper_motors/unipolar_bipolar/). [Přístup získán 15. 4. 2014].
- [12] *MS Struktury s krokovými motory*, VŠB TU Ostrava.
- [13] „MikroElektronika,“ [Online]. Available: [http://www.mikroe.com/downloads/get/1220/stepper\\_manual\\_v100.pdf](http://www.mikroe.com/downloads/get/1220/stepper_manual_v100.pdf). [Přístup získán 16. 2. 2014].
- [14] „Stepper Motor Basics,“ [Online]. Available: <http://www.solarbotics.net/library/pdflib/pdf/motorbas.pdf>. [Přístup získán 2. 3. 2014].
- [15] H. Max, „STEPPER MOTOR DRIVING,“ [Online]. Available: [http://users.ece.utexas.edu/~valvano/Datasheets/Stepper\\_ST.pdf](http://users.ece.utexas.edu/~valvano/Datasheets/Stepper_ST.pdf). [Přístup získán 16. 1. 2014].
- [16] I. K. Polák, „Sběrnice CAN,“ 16. 6. 2003. [Online]. Available: <http://www.elektrorevue.cz/clanky/03021/index.html>. [Přístup získán 12. 2. 2014].
- [17] T. Kugelstadt, „The RS-485 Design Guide,“ 2008. [Online]. Available: <http://www.ti.com/lit/an/slla272b/slla272b.pdf>. [Přístup získán 3. 15. 2014].
- [18] „ATMEL,“ [Online]. Available: <http://www.atmel.com/>. [Přístup získán 23. 1. 2014].
- [19] „MICROCON,“ [Online]. Available: <http://www.microcon.cz/>. [Přístup získán 22. 12. 2013].
- [20] „RMS Technologies,“ [Online]. Available: [http://www.rmsmotion.com/ecart/view\\_category.aspx](http://www.rmsmotion.com/ecart/view_category.aspx). [Přístup získán 22. 1. 2014].
- [21] „FTDI,“ [Online]. Available: <http://www.ftdichip.com/EVE.htm>. [Přístup získán 15. 3. 2014].

- [22] „QVGA-TFT-PCAP-Modul Datenblatt,“ [Online]. Available: <http://www.hy-line.de/fileadmin/hy-line/computer/csv/datasheets/SCF0305434GGC00.pdf>. [Přístup získán 16. 2. 2014].
- [23] H. Tanaka, „Ferroelectric Memory Technology Challenges,“ 9. 8. 2008. [Online]. Available: <http://dtfl.snu.ac.kr/research/fram/fram.htm>. [Přístup získán 12. 4. 2014].
- [24] „Paměť FRAM,“ [Online]. Available: [http://cs.wikipedia.org/wiki/Pam%C4%9B%C5%A5\\_FRAM](http://cs.wikipedia.org/wiki/Pam%C4%9B%C5%A5_FRAM). [Přístup získán 12. 4. 2014].
- [25] „Inter-Integrated Circuit (I2C) Module - SPRUE11C,“ 2005. [Online]. Available: <http://www.ti.com/lit/ug/sprue11c/sprue11c.pdf>. [Přístup získán 8. 2. 2014].
- [26] „Krabíčka HAMMOND,“ [Online]. Available: <http://www.gme.cz/img/cache/doc/622/976/hlinikova-krabicka-u-ha1455b1002-datasheet-1.pdf>. [Přístup získán 16. 4. 2014].
- [27] „Krokové motory - všeobecné údaje,“ 8. 9. 2013. [Online]. Available: [http://www.regulace.cz/CZ/KM/km\\_vu.html](http://www.regulace.cz/CZ/KM/km_vu.html). [Přístup získán 20. 12. 2013].
- [28] I. P. Rydlo, „KROKOVÉ MOTORY A JEJICH ŘÍZENÍ,“ 2000. [Online]. Available: <http://www.mti.tul.cz/files/ats/krok2.pdf>. [Přístup získán 3. 12. 2014].
- [29] „1-3-5 Stepping Motor,“ [Online]. Available: <http://www.nidec.com/en-NA/technology/motor/basic/00008/>. [Přístup získán 15. 2. 2014].
- [30] „SPI komunikace,“ [Online]. Available: [http://upload.wikimedia.org/wikipedia/commons/thumb/b/b6/SPI\\_timing\\_diagram.svg/1000px-SPI\\_timing\\_diagram.svg.png](http://upload.wikimedia.org/wikipedia/commons/thumb/b/b6/SPI_timing_diagram.svg/1000px-SPI_timing_diagram.svg.png). [Přístup získán 22. 12. 2013].

## Slovník pojmů

- **Úhel kroku** je jmenovitý úhel, o který se otočí hřídel motoru na jeden řídicí impuls.
- **Řídicí kmitočet** je kmitočet, kterým je řízen motor. Souhlasí s kmitočtem kroku, jestliže motor běží bez chyby kroku.
- **Magnetická klidová poloha** je poloha, kterou zaujme rotor vybuzeného motoru, jestliže chyba statistického úhlu se rovná nule.
- **Tolerance úhlu kroku** je největší kladná nebo záporná statická odchylka úhlu vůči jmenovitému úhlu kroku. Tato odchylka může nastat, jestliže se rotor motoru pootočí o 1 krok z jedné magnetické polohy do druhé. Přitom se vychází ze vztažné magnetické klidové polohy. Měří se v průběhu celé otáčky rotoru.
- **Vlastní přídržný moment** je maximální moment, kterým může být staticky zatížena hřídel nevybuzeného motoru, aniž by se začala plynule otáčet.
- **Přídržný moment** je maximální moment, kterým může být staticky zatížena hřídel vybuzeného motoru, aniž by se začala plynule otáčet.
- **Statický zatěžovací úhel** je úhel, o který se otočí hřídel motoru zatížená předem daným statickým momentem vůči nezatíženému stavu (tj. magnetické klidové poloze) při nulové řídicí frekvenci.
- **Zatěžovací moment** je moment charakteru pasivního tření, kterým je zatížena hřídel motoru.
- **Stabilizační zatěžovací moment** je zatěžovací moment nutný pro funkci krokového motoru v nestabilní oblasti řídicího kmitočtu.
- **Nestabilní oblast řídicího kmitočtu** je část charakteristiky rozběhového momentu a rozběhového momentu setrvačnosti, která má bez stabilizačního zatěžovacího momentu inflexní bod.
- **Rozběhový moment** je zatěžovací moment, se kterým se motor může rozběhnout start-stop bez chyby kroku, bez přidavné vnější setrvačné hmoty při definovaném řídicím kmitočtu.
- **Rozběhový moment setrvačnosti** je vnější moment setrvačnosti, se kterým se motor může rozběhnout start-stop bez chyby kroku, bez zatížení zatěžovacím momentem při definovaném řídicím kmitočtu.
- **Chod naprázdno** je provozní stav, ve kterém motor není zatížen ani vnějším zatěžovacím momentem ani vnější setrvačnou hmotou.
- **Maximální rozběhový kmitočet** je největší řídicí kmitočet, při kterém se motor může rozběhnout start-stop při chodu naprázdno bez ztráty kroku.
- **Rozběhový kmitočet** je největší řídicí kmitočet, při kterém se může motor rozběhnout bez ztráty kroku s určitým zatížením sestávajícím z rozběhového momentu a z vnějšího momentu setrvačnosti.
- **Maximální provozní kmitočet** je nejvyšší řídicí kmitočet, při kterém může být motor provozován při plynulém zvýšení nebo snížení řídicího kmitočtu při chodu naprázdno. [27]

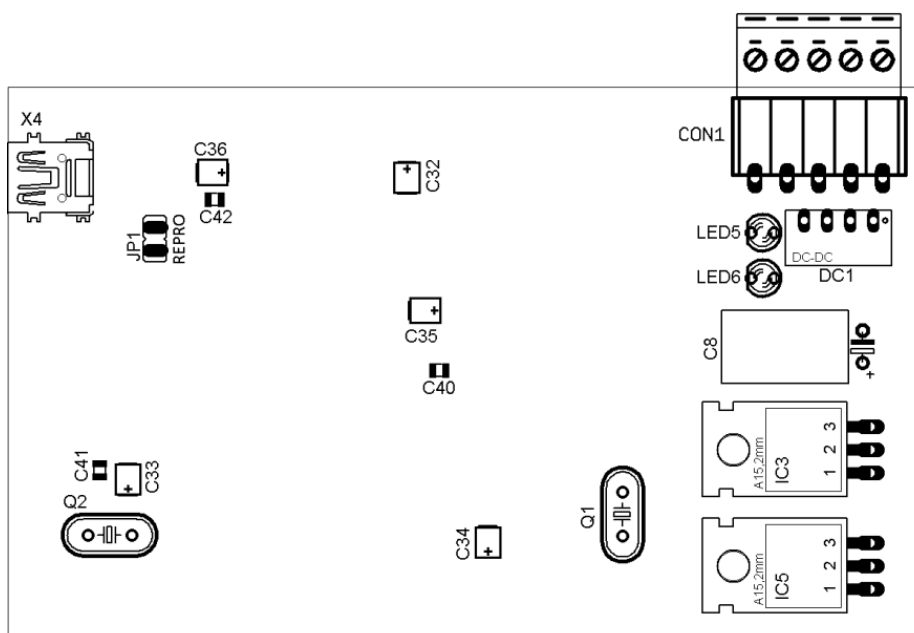


### **Seznam příloh:**

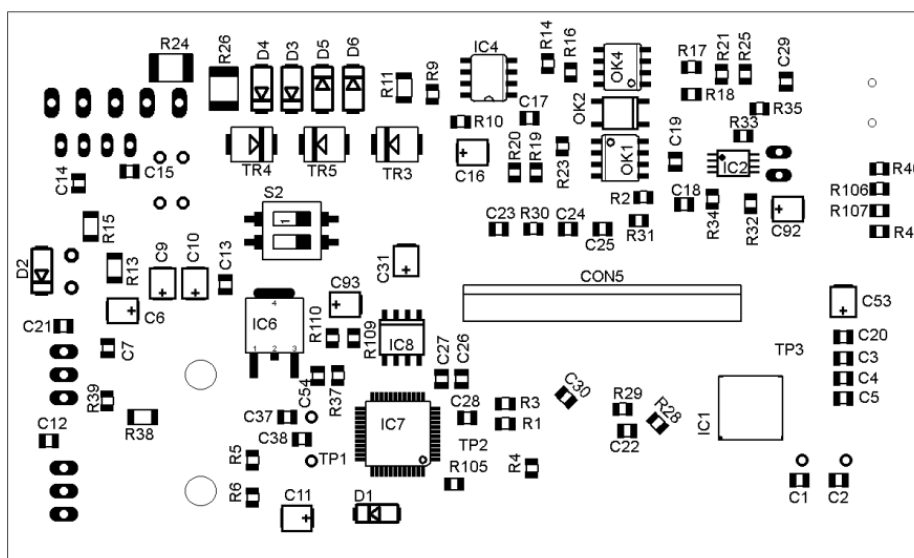
- Příloha 1: Schéma zapojení
- Příloha 2: Deska plošných spojů - osazovací plány
- Příloha 3: Deska plošných spojů – seznam součástek
- Příloha 4: Deska plošných spojů – vygenerovaná data
- Příloha 5: Program – seznam hlavních podprogramů 1
- Příloha 6: Program – seznam hlavních podprogramů 2



## Deska plošných spojů - osazovací plány



Obrázek 1 Osazovací plán vrchní strany desky

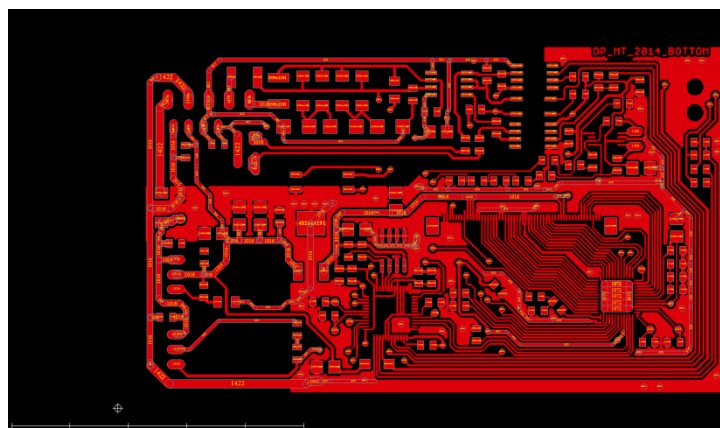


Obrázek 2 Osazovací plán spodní strany

## Seznam součástek

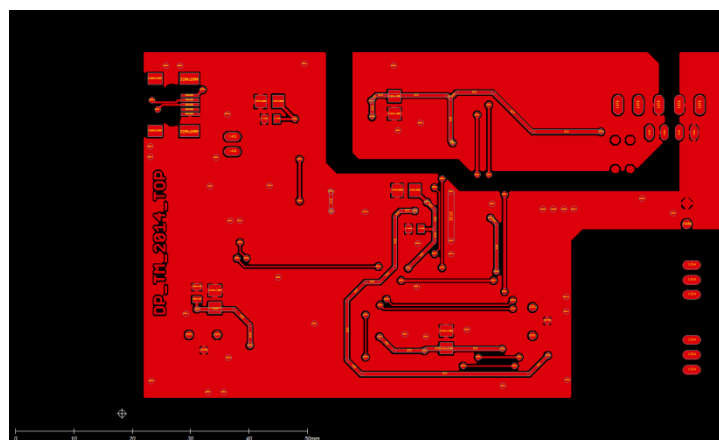
Pozice	Hodnota	Pouzdro	Pozice	Hodnota	Pouzdro	Pozice	Hodnota	Pouzdro
C1	22p	SMD 0805	C35	33uF/10V	3528	R3	100k	SMD 0805
C2	22p	SMD 0805	C36	33uF/10V	3528	R4	100k	SMD 0805
C3	100n	SMD 0805	C37	15p	SMD 0805	R7	4k7	SMD 1206
C4	100n	SMD 0805	C38	15p	SMD 0805	R8	49,9R	SMD 1206
C5	100n	SMD 0805	C39	33uF/10V	3528	R9	47k	SMD 0805
C6	33uF/10V	3528	C40	100n	SMD 0805	R10	47k	SMD 0805
C7	100n	SMD 0805	C41	100n	SMD 0805	R11	150R	SMD 1206
C8	E100M/50VT	RM35/10	C42	100n	SMD 0805	R12	4k7	SMD 1206
C9	33uF/10V	3528	C43	100n	SMD 0805	R14	10k	SMD 0805
C10	33uF/10V	3528	C44	2n2	SMD 0805	R16	22k	SMD 0805
C11	33uF/10V	3528	C45	220n	SMD 0805	R17	1k	SMD 0805
C12	100n	SMD 0805	C53	33uF/10V	3528	R21	10k	SMD 0805
C13	100n	SMD 0805	C54	100n	SMD 0805	R24	140mA	polysw-1812
C14	100n	SMD 0805	C92	33uF/10V	3528	R25	1k	SMD 0805
C15	100n	SMD 0805	C93	33uF/10V	3528	R26	140mA	polysw-1812
C16	33uF/10V	3528	CON1	MRT35	konektor	R28	1k	SMD 0805
C17	100n	SMD 0805	CON5	FPC_54	konektor	R29	1k	SMD 0805
C18	470n	SMD 0805	D1	1N4007	DO214	R30	1k	SMD 0805
C19	100n	SMD 0805	D2	1N4007	DO214	R31	10k	SMD 0805
C20	100n	SMD 0805	D3	1N4148	DO35-7	R32	22k	SMD 0805
C21	100n	SMD 0805	IC1	FT800A	VQFN48	R33	22k	SMD 0805
C22	4n7	SMD 0805	IC2	TPA2005D1	DGN	R34	10k	SMD 0805
C23	4n7	SMD 0805	IC3	LM317HVT	TO220	R35	47k	SMD 0805
C24	4n7	SMD 0805	IC4	75176	SO08	R37	100k	SMD 0805
C25	470n	SMD 0805	IC6	LF33CDT	TO252	R38	4k7	SMD 1206
C26	100n	SMD 0805	IC7	AT32UC3B1256	TQFP48	R39	1k8	SMD 0805
C27	100n	SMD 0805	IC8	FM24LC64	SO-08	R105	10k	SMD 0805
C28	100n	SMD 0805	IC9	MIC2289A	MSOP08	R106	49,9	SMD 0805
C29	100n	SMD 0805	JP1	JP1E	Kon. Rep.	R107	49,9	SMD 0805
C30	1n	SMD 0805	L1	NR3015T220M	SMC 12	R109	10k	SMD 0805
C31	33uF/10V	3528	Q1	16MHz	HC49U-V	R110	10k	SMD 0805
C32	33uF/10V	3528	Q2	12MHz	HC49U-V	S2	SWS002	SMS-002
C33	33uF/10V	3528	R1	100k	SMD 0805	TR3	P6SMB10A	SMB
C34	33uF/10V	3528	R2	1k	SMD 0805	TR4	P6SMB10A	SMB
šroubky			chladič			X4	MINI_USB	SMD
			DO2					

### Deska plošných spojů - vygenerovaná data



*Obrázek 3 Vodivé spoje vrstvy "BOTTOM"*

### Deska plošných spojů - vygenerovaná data vrstva TOP



*Obrázek 4 Vodivé spoje vrstvy "TOP"*

**Program – seznam hlavních podprogramů 1****main.c // Hlavní program – hlavní smyčka**

```
1 int main (void);
```

**init.c // Základní nastavení procesoru, portů , komunikací**

```
1 void main_init_proc (void);
2 void init_usart(void);
3 void spi_port_init(void);
4 void NumToUart16(U16 Num);
5 void NumToUart32(U32 Num);
6 void Set_Baudrate_RS485(unsigned char Baudrate_RS485);
7 void read_eeprom(void);
```

**ft800.c // Nastavení displeje a rutiny pro odesílání (čtení) dat na adresy přes SPI**

```
1 void Delay1(unsigned short int Time);
2 void Ft800_init(void);
3 void FT800_SPI_HostCommandDummyRead(void);
4 void FT800_SPI_HostCommand(unsigned char Host_Command);
5 void FT800_SPI_SendAddressRD(U32 Memory_Address);
6 void FT800_SPI_Write16(U16 SPIValue16);
7 void FT800_SPI_Write32(U32 SPIValue32);
8 void FT800_SPI_SendAddressWR(U32 Memory_Address);
9 void FT800_SPI_send_data8(U32 Address, unsigned char Data);
10 void FT800_SPI_send_data16(U32 Address, U16 Data);
11 void FT800_SPI_send_data32(U32 Address, U32 Data);
12 void FT800_SPI_send_data_n(U32 Address, U16 n, unsigned char *Data);
13 U32 FT800_SPI_Read32(U32 Address);
14 void FT800_Play_Sound(unsigned char Midi_Note, unsigned char Effect, unsigned char Volume);
15 unsigned char FT800_SPI_Read8(U32 Address);
16 U16 FT800_SPI_Read16(U32 Address);
```

**iic.c // Komunikace přes IIC sběrnici – obsluha paměti**

```
1 void Wait_IIC(void);
2 void Start(void);
3 void Stop(void);
4 unsigned char WriteI2C(unsigned char Data, unsigned char ack);
5 unsigned char ReadI2C(unsigned char ack);
6 unsigned char Read_Byte_I2C(unsigned int Addr);
7 unsigned char Write_Byte_IIC(unsigned int Addr, unsigned char Data);
8 unsigned char Write_U32_IIC(unsigned int Addr, unsigned long int Data);
9 unsigned long int Read_U32_IIC(unsigned int Addr);
10 int eeprom_read_bytes(U16 eeaddr, char * buf, int len);
11 int eeprom_write_bytes(U16 eeaddr, unsigned char * buf, int len);
12 unsigned char Write_Int_IIC(unsigned int Addr, unsigned int Data);
13 unsigned int Read_Int_IIC(unsigned int Addr);
```

**Program – seznam hlavních podprogramů 2****t\_screen.c // Obsluha dotykového panelu**

```

1 void ft800_Touch_XY(void);
2 unsigned char touch_x_5 (U16 position);
3 void set_beep(unsigned char tone);
4 void Test_Top_Line_Touchscreen (U16 pos_x);
5 unsigned char Test_Low_Line_Touchscreen (U16 pos_x);
6 unsigned char ft800_Touch_OK(void)

```

**displays.c // Displej – kreslení obrázků, klávesnice a tlačítek**

```

1 U32 TopLine(U32 cmdOffset, unsigned char choose);
2 U32 Text_To_Disp(U32 cmdOffset,U16 pos_x,U16(pos_y),unsigned char opt,const char *Num_string);
3 U32 keyboard(U32 cmdOffset);
4 void display_01(void);
5 void set_topline_value (unsigned char tp_lin);
6 U32 CommandLine(U32 cmdOffset);
7 unsigned char command_line_set_char(unsigned char num);
8 unsigned char command_line_clear_char(void);
9 void set_keyboard_type(void);
10 U32 SmallPositionLine(U32 cmdOffset,unsigned char offset_position);
11 unsigned char send_ASCII_string(void);
12 U32 Arrow_Send_Rec (U32 cmdOffset, unsigned char Vys_prijem);
13 void display_02(void);
14 U32 Send_Rs485_Text(U32 cmdOffset,unsigned char flag_receive);
15 U32 Rec_command(U32 cmdOffset);
16 U32 query_command(U32 cmdOffset);
17 unsigned char cursor_plus_minus (unsigned char plus_minus);

```

**gpio.c // Obsluha portů – knihovna výrobce, zobrazena jen použitá část**

```

1 void gpio_set_gpio_pin(unsigned int pin);
2 void gpio_clr_gpio_pin(unsigned int pin);
3 void gpio_tgl_gpio_pin(unsigned int pin);
4 int gpio_enable_module_pin(unsigned int pin, unsigned int function);
5 void gpio_enable_gpio(const gpio_map_t gpiomap, unsigned int size);

```